3-31-2010

ENHANCEMENT OF CHURN PREDICTION ALGORITHMS

Matthew N. Anyanwu

To the University Council:

The dissertation Committee for Matthew Nwokejizie Anyanwu certifies that this is the approved version of the following dissertation:

Enhancement of Churn Prediction Algorithms

Sajjan Shiva, Ph.D., Major Professor

Vasile Rus, Ph.D.

Qishi Wu, Ph.D.

Ebenezer George, Ph.D.

Accepted for the Council:

Karen D. Weddle-West, Ph.D.
Vice Provost for Graduate Programs

ENHANCEMENT OF CHURN PREDICTION ALGORITHMS

by

Matthew Nwokejizie Anyanwu

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

Major: Computer Science

The University of Memphis

August, 2010

Dedication

This study is dedicated to my darling wife and son:

Oluchukwu and Oluebubechukwu Anyanwu

Acknowledgements

I am grateful to Almighty God for his guidance and infinity mercy.

I would like to thank my academic adviser, Dr. Sajjan Shiva, for his support and continuous encouragement throughout this study.

I would also like to thank my dissertation committee members: Dr.Vasile Rus, Dr. Qishi Wu, and Dr. Ebenezer George for their encouragement, advice, support, comments, suggestions, and constructive criticism. I would also like to thank Dr. Dipankar Dasgupta for his advise, suggestions, and financial support of my research work.

I am also grateful to the graduate school at the University of Memphis for awarding me the prestigious "Van Vleet Memorial Doctoral Award" for a period of four years. This prestigious prize enabled me to complete my doctoral program. Very special thanks goes to my immediate family: my wife Oluchukwu and my son Oluebubechukwu for their encouragement, understanding, patience, and support throughout the period of the dissertation research and my academic program in general.

Abstract

Anyanwu, Matthew N. Ph.D. The University of Memphis. August 2010. Enhancement of Churn Prediction Algorithms. Major Professor: Sajjan Shiva, Ph.D.

Customer churn can be described as the process by which consumers of goods and services discontinue the consumption of a product or service and switch over to a competitor. It is of great concern to many companies. Thus, decision support systems are needed to overcome this pressing issue and ensure good return on investments for organizations. Decision support systems use analytical models to provide the needed intelligence to analyze an integrated customer record database to predict customers that will churn and offer recommendations that will prevent them from churning [32]. Customer churn prediction, unlike most conventional business intelligence techniques, deals with customer demographics, net worth-value, and market opportunities. It is used in determining customers who are likely to churn, those likely to remain loyal to the organization, and for prediction of future churn rates. Customer defection is naturally a slow rate event, and it is not easily detected by most business intelligent solutions available in the market; especially when data is skewed, large, and distinct. Thus, accurate and precise prediction methods are needed to detect the churning trend. In this study, a churn model that applies business intelligence techniques to detect the possibility that a customer will churn using churn trend analysis of customer records is proposed. The model applies clustering algorithms and enhanced SPRINT decision tree algorithms to explore customer record database, and identify the customer profile and behavior patterns. The Model then predicts the possibility that a customer will churn. Additionally,

it offers solutions for retaining customers and making them loyal to a business entity by recommending customer-relationship management measures.

Contents

List of Tables **x**

List of Figures **xii**

1 Introduction **1**

1.1 Goals . . . 3

1.2 Main Contributions . . . 5

1.3 Dissertation Outline . . . 6

2 Background **8**

2.1 Decision Support Systems (DSS) . . . 8

2.1.1 Expert Systems . . . 12

2.1.2 Knowledge Management System . . . 13

2.1.3 Business Intelligence . . . 14

2.1.4 Application of BI to Customer Churn Analysis . . . 21

2.1.5 Data Warehousing . . . 24

2.1.6 Data Mining . . . 28

3 Clustering Algorithm **33**

3.1 Cluster Validity . . . 35

3.2 Cluster Validity Measures . . . 36

3.2.1 Internal Criteria Measures . . . 37

3.2.2 External Criteria Measures . . . 42

3.2.3 Relative Criteria Measures . . . 47

3.3 Validity Measures Statistics . . . 48

3.4 Experimental Analysis . . . 53

3.4.1 Data Set Usage . . . 55

3.4.2 Experimental Results . . . 56

3.4.3 Test Observation 57
3.5 Summary 59

4 Classification and Prediction Algorithm 60
4.1 Classification Algorithm 61
4.1.1 Classification Accuracy 62
4.2 Decision Tree Algorithm 64
4.2.1 Tree Growth 66
4.2.2 Tree Pruning 70
4.3 Serial Implementation of Decision Tree Algorithm 72
4.3.1 IDE3 74
4.3.2 C4.5 75
4.3.3 CART 75
4.3.4 SLIQ 76
4.3.5 SPRINT 77
4.4 Serial Decision Tree Algorithm Implementation Statistics 79
4.5 Experimental Analysis 79
4.5.1 Experimental Results 80
4.6 Predicting Model 86
4.6.1 Predicting Model Complexity: 86
4.6.2 Model Evaluation: 88
4.7 Summary 88

5 Churn Analysis and Management 91
5.1 Churn Analysis 92
5.1.1 Churn Indices 94
5.1.2 Churn Management 95
5.1.3 Churn Model 96

6 Implementation and Results 99
6.1 Proposed Algorithm 99
6.2 Algorithm Complexity of Decision Tree 104
6.3 Implementation 105
6.3.1 Data Preparation and Algorithm Processing 106
6.3.2 Implementation The Proposed Model using Weka 110

6.3.3 Implementation Results 115

7 Discussion and Conclusions **122**

7.1 Main Contributions 125

7.2 Future Work 126

Bibliography **127**

Appendix **139**

List of Tables

3.1 Data Partition Set 43
3.2 Literature review of cluster validity measures 49
3.3 Frequency use of each measure 50
3.4 Summary of Internal Measure Statistics 51
3.5 Summary of External Measure Statistics 52
3.6 Euclidean Distance Measurement 57
3.7 Cluster and Data set Correlation Measurement 57

4.1 Classification Matrix 63
4.2 Literature review of Decision Tree Algorithms 80
4.3 Frequency use of Decision Tree Algorithm 81
4.4 Statlog Datasets 83
4.5 Execution Time to build Model 83
4.6 Classification Accuracy 83

6.1 Retail Data 107
6.2 Financial (bank) data 107
6.3 Medical (Heart Disease) data 108
6.4 Churn prediction of retail data 108
6.5 Churn prediction of financial data 109
6.6 Churn prediction for heart disease 109
6.7 Comparative analysis of classification accuracy using retail data 117
6.8 Comparative analysis of classification accuracy using Medical (Heart disease) data 117
6.9 Comparative analysis of classification accuracy using Financial Data 118
6.10 Comparative analysis for positive bike prediction 118
6.11 Comparative analysis for negative bike prediction 118

6.12 Comparative analysis for Absence of Heart Disease prediction 119
6.13 Comparative analysis for Presence of Heart Disease prediction 119

List of Figures

2.1 Decision-Making Process Steps [123] 10
2.2 BI Creation and Components [134] 16
2.3 Data Warehouse Architecture [18] 27
2.4 Regression for Debt-Income Data set [31] 32

3.1 The three steps of clustering 34
3.2 Clustering of Loan Data set into three clusters[31] 35
3.3 Internal and External Measure Frequency Usage 50
3.4 Validity Measures Frequency Usage 52
3.5 Developed Cluster Validity System 54

4.1 Decision Tree Classification Algorithm 66
4.2 Decision Tree Building Phase 68
4.3 Execution time and Number of Records 84
4.4 Execution time and Number of Attributes 85

5.1 Framework of churn prediction and management 96

6.1 The proposed Model 102
6.2 The proposed Model Framework [50] 103
6.3 Bike Buyer Class Prediction 120
6.4 Heart Disease Prediction Decision Tree 121

Chapter 1

Introduction

Customer churn is the loss of clients to business competitors. It occurs because of keen competition in the market environment. Thus organizations need to adopt strategies that will win their customers over and make them loyal. In this study we propose a decision support model which uses analytical models to provide the needed intelligence to analyze an integrated customer record database to predict customers that will churn and offer recommendations that will prevent them from churning as the cost of winning new customers is far more than the cost of retaining them [32]. Also, it is very difficult and tasking to win back the customers that have been lost to a competitor. Most business intelligence solutions available in the market (see sections 2.1.4.1 and 2.1.3.2) were reviewed and analyzed, then it was discovered that they are not well suited for churn prediction, especially when the data is skewed, large, and distinct. Also, churning generally is of low occurrence in nature which makes it difficult for most Business Intelligence(BI) solutions to detect it.

The proposed model builds a churn algorithm that utilizes the database back-end and Structured Query Language (SQL) query rules to explore and analyze customer data records. The model first integrates all the data used for business activity into one common source using data warehousing techniques, and performs churn trend analysis on the customer records to produce customer profile and behavior patterns. Enhanced Data mining techniques are then applied to predict the likelihood that a customer will churn and offer solutions on how to manage the customers, which will lead to a high return on investment to a business entity. The proposed churn model includes business intelligence techniques, data warehousing techniques, clustering, and enhanced decision tree algorithms, among others. The functionalities of a SPRINT decision tree algorithm, which are scalability and fastness in data modeling are exploited. Also, the SPRINT algorithm is enhanced to enable it to detect churn trend movement by improving its attribute splitting ability. Getting organizations to release their transaction data in testing our proposed model was difficult. We used data from Microsoft retail data mining records (Table 6.1); financial bank data records from De Paul University, data resource (Table 6.2) as described in section 6.3.1; medical data records from V.A. Medical Center, Long Beach; Cleveland Clinic Foundation (Table 6.3); [26] and some randomly generated data.

The validation of the model is in two phases. In the first phase we developed a system that validate the result of the clustering algorithms using different validity measures. A literature survey was conducted on all the available validity measures. The experimental analysis of cluster validity measures and survey analysis showed that Dunn's cluster validity measure In-

dex is preferred among the internal measures, while the Rand cluster validity measure Index is preferred among the external measures, but Dunn's Index is the preferred cluster validity measure when both internal and external cluster validity measures are compared. In the second phase of the model validation process, the data sets was divided into training and test sets. The test data sets are used in validating the result of the enhanced decision tree algorithm. Experiments on how to determine the performance of our proposed algorithms when compared with other common decision tree algorithms was conducted. A comparative analysis of our proposed algorithm with other decision tree algorithms with respect to customer churn prediction using the retail, medical and financial bank data sets shows a classification accuracy above 90% and a good improvement of other classification accuracy measures. Thus the proposed algorithm is suitable for customer churn prediction.

1.1 Goals

The aim of this study is to contribute to the on-going research of building a comprehensive churn algorithm that will use the required business intelligence techniques to analyze a customer record in a central repository. The model will specifically predict the possibility that a client will churn, especially when the data is skewed, large, and distinct. The data features (skewness, size and distinctness) have not been well addressed by other business intelligence solutions available in the market. The model will also, forecast future churn rate and offer

customer relationship management measures that will address churning trend. The objectives of the research are as follows:

- Collect and consolidate customer data record of the target organization and from external sources;
- Create a data warehouse/data mart/database of all the data collected;
- Perform market analysis to determine high value and low value customers, and basket analysis to show products that are requested together;
- Develop a system that compares the common cluster validation algorithms;
- Conduct a literature survey that compares all the cluster validation algorithms;
- Select the best clustering validation algorithm from the literature survey and the clustering validation system;
- Use the validation algorithm to validate the clustered data;
- Apply the enhanced proposed model to predict customers that will churn;
- Use the identified pattern to predict, churn trend analysis, customer churn profiling and future churn rate;
- Suggest customer relationship management technique that will make the customer loyal.

1.2 Main Contributions

The main objective of this dissertation research as stated in section 1.1 is to apply business intelligence techniques to predict the possibility that a customer will churn. Thus, this study offers suggestions and enhancements on how to deal with the identified issues. In this study, a combination of data mining techniques, including clustering and decision tree, were used in designing the proposed churn model. SPRINT decision tree algorithm as it is implemented cannot accurately predict the possibility that a customer will churn. However, it is enhanced by changing the splitting attribute from gini/entropy to gain ratio. Also, clustering algorithm is first applied to the data set in-order to determine churn trend analysis and customer profile that will be used both in the churn prediction and management. SPRINT decision tree algorithm is preferred among other classification/prediction algorithms because of the following features:

- Decision tree algorithm is an eager learning algorithm;
- Decision tree algorithm is easy to understand and implement;
- It is based on rules, thus making it easy to leverage on database back-end engine like SQL queries, which make the use of database relational management system (Microsoft SQL server, Oracle etc.) possible;
- No external information is required in building the model apart from the one in the training data set;

- SPRINT algorithm is fast, scalable and disk oriented;
- SPRINT algorithm can be implemented in serial or parallel pattern;
- Data items are only sorted once at the nodes during tree building phase in SPRINT algorithm; and
- SPRINT algorithm handles both continuous and categorical attributes.

The clustering process is used to identify the natural organization and patterns in the data by discovering similarities and differences and deriving useful conclusions [138]. Also, a cluster validation system that validates the common cluster validation algorithms is designed to validate and identify a good validation algorithm.

1.3 Dissertation Outline

Chapter 2 provides a detailed background on decision support system (DSS) applications and usage, especially its application to customer churn prediction and management. A review of business intelligence techniques and different styles of business intelligence systems and their applications to churn management are discussed. Finally, a review of data mining techniques and the process of building a data warehouse for DSS is described.

Chapter 3 provides a review of clustering algorithm, clustering validation method, and a system that compares clustering validation methods. The system is used to identify the best

clustering validation algorithm that is used in validating the result of experimental analysis in predicting the possibility that a customer will churn.

Chapter 4 describes the classification algorithms. Performance measures used to determine the accuracy of classification algorithms are reviewed. A detailed review of decision tree algorithms, including all the phases of decision tree construction are provided.

In chapter 5, churn prediction and management model is presented. The churn indices that determine when a client churns are described, also the problems identified with each churn model is explained.

In chapter 6, the proposed churn model is presented. The proposed model uses business intelligence methods to detect the possibility that a customer will churn. This will enable management to be proactive in making strategic decisions to reduce the rate at which customers churn. Experimental analysis is used to determine the performance and accuracy of the proposed algorithm. Furthermore the enhanced Sprint decision tree algorithm with other commonly used decision tree algorithms are experimentally compared. The experimental analysis shows that our algorithm has 90% classification/prediction accuracy in predicting the possibility that a customer will churn.

Finally, Chapter 7 presents the conclusions and suggestions for future research work.

Chapter 2

Background

2.1 Decision Support Systems (DSS)

A DSS is an interactive (human component) and computer based application used in analyzing, supporting complex decisions, and resolving problems that would otherwise be solved by humans [118]. Thus, a DSS system includes both the human and computer-based technology components. Most decision problems are so complex and intricate to solve that they require both human and computer-based approaches to find solutions to them. However, DSS supports the human cognitive aspect in arriving at a timely solution to complex decision problems. DSS can equally be described as systems and subsystems that help decision makers in organizations to use information technologies, data, models, and documents in choosing the best result among multiple alternatives [98, 27]. It is used in making operational, analytical, and strategic decisions [35]. DSS collects data from legacy systems, external and relational

systems (database and data warehouse) analyses, and then processes the data items effectively and efficiently. In order to offer alternative solutions based on processing the data, the human component of the DSS reviews and selects the best solution to solve problems or make the best decisions [98, 131]. The human component of DSS is extremely essential because the computer based component only complements humans in dealing with complex decision-making. Thus, all DSS applications need user-interfaces that will determine the suitability of the DSS [27]. Therefore, decision making processes involve the following steps [123]:

- Intelligence Step: This step involves the computer and human components of DSS. In this stage the opportunities or threats (problems) in the environment are identified and analyzed.

- Design Step: Alternative solutions are preferred to the complex decision problems. The human and computer components are also involved in this stage.

- Choice Step: A choice is made from the alternative solutions suggested in the design stage. The computer and human components of DSS are involved in selecting the best choice for a particular decision problem.

- Implementation Step: When a particular choice among possible alternatives are made, then its successful implementation is monitored to ensure that the desired result is obtained.

Figure (2.1) shows the Decision-making process steps.

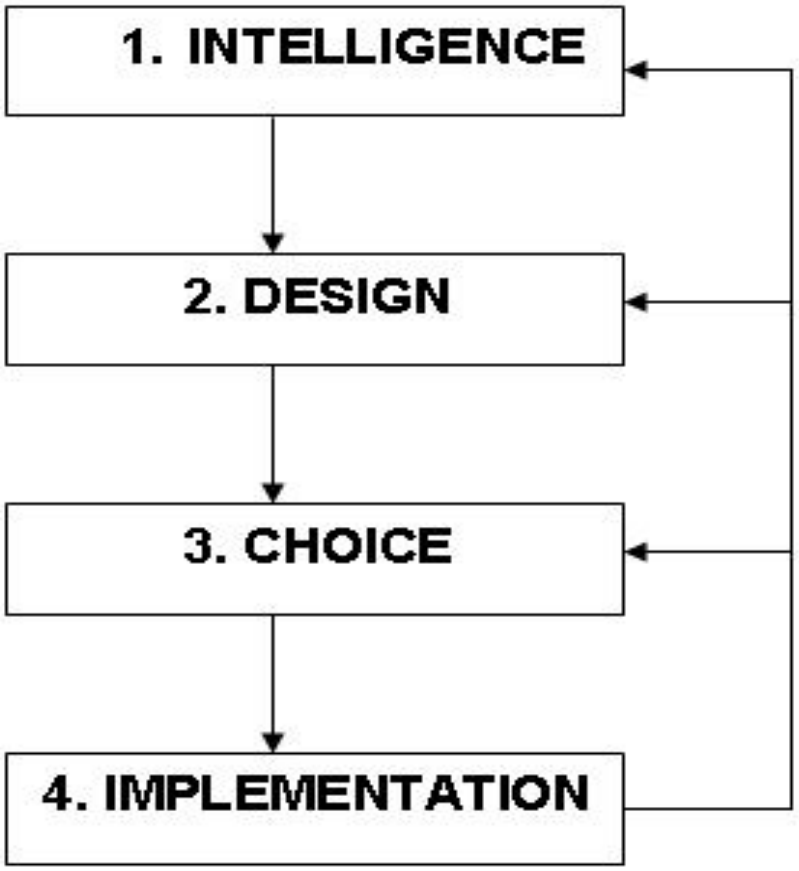

Figure 2.1: Decision-Making Process Steps [123]

A critical survey of DSS articles and journals [98, 131, 27] shows that DSS applications are mainly used for manipulation of qualitative and quantitative models access, retrieval, and analysis. They are also used in the processing of data items in databases and data warehouses, decision-making, and for collaboration and cooperation between systems and users [98]. The reviews classify DSS applications into taxonomies based on the assistance criterion of

solving decision problems [98]. The taxonomies of the DSS based on the reviews are stated as follows:

- Model-Driven DSS: Model-Driven DSS is used in manipulating qualitative and quantitative models such as scheduling, decision analysis, financial, and statistical, simulations, and others. These models are used in solving complex decision problems.

- Data-Driven DSS: In this approach, data from legacy systems, operational systems and external systems are collected and collated into databases and data warehouses. The databases and data warehouses are then queried to obtain specific results that will be used in solving decision problems.

- Document-Driven DSS: This approach emphasizes the use of documents of various data formats such as database of customer records, text documents, and multimedia documents that are stored in computer systems to solve decision problems. The fast and timely retrieval of these documents and its processing aids in arriving at solutions to decision problems.

- Communication-Driven DSS: This approach is used for intra-communication among employees in an organization or for users of DSS application to collaborate and communicate. It can take place in an office, home, or on the Web (e.g., instant messaging, chatting, net/online conference meeting, groupware, video conferencing and bulletin boards). It is normally developed using network technologies like the web or client-server based architecture; thus, the users of DSS are in a virtual environment.

- Knowledge-Driven DSS: In this approach, computer-based systems with specialized rules or skills in a particular domain are used to suggest solutions to problems. These types of DSS applications act as an expert system discussed in section 2.1.1.

DSS includes, but it is not limited to, knowledge management systems, business-intelligence systems and expert systems [98, 134]. It shows the way these technologies can be applied in decision-making individuals who are professionals, such as business executives or managers in order to shape the world, their personal lives, and the business environment [111].

2.1.1 Expert Systems

Expert Systems are part of Artificial Intelligence Systems, including computer-based systems with human intelligence that can be used in decision-making, as well as reasoning and problem solving [111]. Hence they act as DSS. Expert system is a computer-based system with an application software that reproduces the performance of human intelligence, including cognitive skills. Cognitive skills include the ability to understand, learn, and provide solutions to decision problems in a particular domain or field of study [83, 2]. Thus, Expert System acts as a consultant to its users in solving a problem, reasoning, learning, or making a complex decision. It provides answers to queries or questions to real-world problems in a particular domain by making inference to human knowledge or intelligence. The inference is made from a knowledge base which contains facts and rules about a particular domain of knowledge and the way the knowledge could be used [111]. Expert systems provide solutions to decision problems, as well as explanations for particular solutions chosen. The process of

building an expert system is called Knowledge Engineering. The basic building blocks of Expert System are the knowledge base and the reference engine (reasoning). The knowledge base contains the actual data and facts about tasks in a particular domain, while the reference engine contains the reasoning process that is used in selecting the right solution to a problem when the Expert System is applied to any given task [83, 2]. Arguably, however, the Expert System is used in solving real problems of which decision making is one of those problems.

2.1.2 Knowledge Management System

The Knowledge Management System is a part of DSS which involves the creation of a knowledge base and the provision of an interface through which the knowledge will be accessed and disseminated [7, 119]. The knowledge base in Knowledge Management includes the Expert System knowledge discussed in section 2.1.1 and the non-expert knowledge with records from text files, pdf files, database/dataware house, and external records. It contains both implicit and explicit knowledge. Knowledge management can also be described as a set of methods or practices that involve the creation and storage of knowledge and its distribution to organizations and individuals that need it. Knowledge management encourages information sharing and the reuse of knowledge [30, 119].

Knowledge has been viewed in various perspectives by many authors [7, 23, 119, 30]. The Oxford English Dictionary defines knowledge as "expertise, and skills acquired by a person through experience or education"[121]. Also, Davenport et al. defines knowledge as "a fluid mix of framed experience, values, contextual information, and expert insights and

grounded intuitions that provides a framework for evaluating and incorporating new experiences and information" [23]. The modern and recent view of knowledge by most business entities is both expert and non- expert based, implicit and non-implicit information/data, data records from data base/datawarehouse, text structured files, pdf files and data in various formats. It includes operation and internal and external data/information. The ability of the Knowledge Management system to reuse and share knowledge makes a good tool in software development, customer-relation management, and other decision-support systems; past mistakes can be avoided, and expertise can be put to good use. Decision-making is a complex and demanding activity which requires the input of the required knowledge to obtain the appropriate solution. DSS applications use Knowledge Management Systems to process and apply the required knowledge to make strategic decisions that will maximize profitability and ensure high returns on investments for organizations [46].

2.1.3 Business Intelligence

Business Intelligence (BI) is a management tool which involves the use of analytical techniques and computer softwares to explore and analyze customer data records in order to produce useful information on organizations' operations and performance indices [113]. It is more like an interdisciplinary field that includes, but is not limited to, data mining, online analytical processing, datawarehouse/database querying and reporting. It also identifies relationships among business entities. BI tools are used by businesses executives to make

key critical decisions on many factors affecting business in order to ensure high returns on investments [113].

The main aim of BI application to customer records is to improve customer-relationship management of organizations which will lead to increased customer base and maintenance of existing customers, further leading to a high return on investments. BI can also be used in determining the possibility that a consumer of a product or service will churn by defecting to other business competitors or discontinuing consumption of products or services. Figure 2.2 shows the creation and components of BI system.

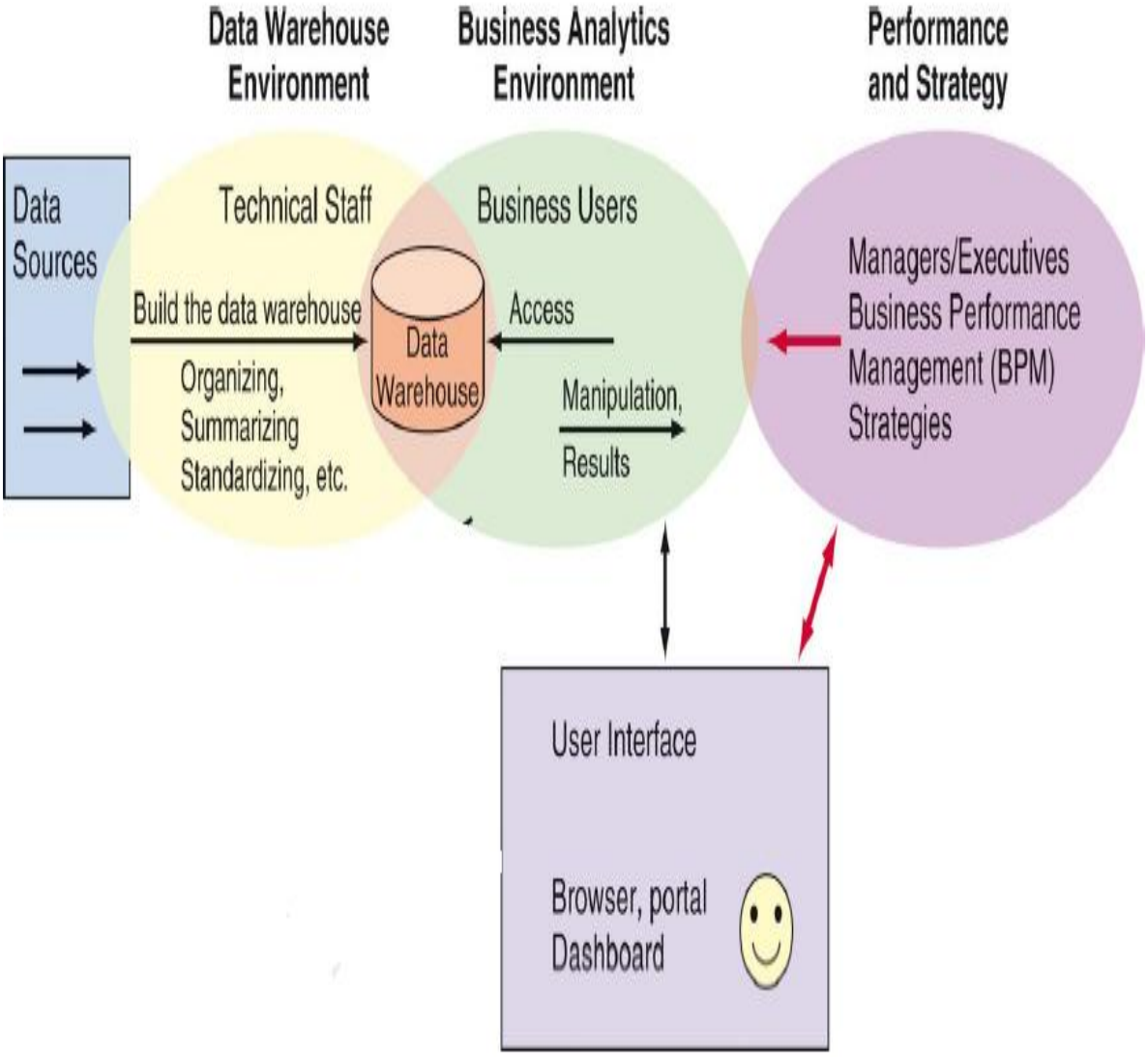

Figure 2.2: BI Creation and Components [134]

2.1.3.1 Application of Business Intelligence Systems

BI can be described as the technology, skills, software applications, and practices that enable business entities to capture, store, analyze, and process data and information to understand their commercial status in order to remain competitive [113]. BI is used in making data easily accessible and facilitates free flow of information within a business entity. BI ensures that the right information is delivered at the right time and to the right users. It also makes sharing of information a possibility within and outside an organization. BI, unlike transactional applications which provide only operational information, provides both analytical and operational that assist the organization in making strategic business decisions that ensure good return on investments [58]. BI tools include both back-end and front-end infrastructure tools that are used in data extraction, capturing, analyzing, reporting, and knowledge extraction that can be applied to decision-making. The back-end tools include database, data warehouse, datamarts, and Online Analytical Processing (OLAP) servers, while front-end tools include portals, scorecards, dashboards, data mining, exploration, modeling, process analytical and decision making tools [76]. In order to use BI tools to increase returns on investments, efficiently and effectively manage a business entity, Lokken [76] , states that BI tools should have the following critical success factors:

- Data Access: BI tools should have access to clean data (because data is the bedrock of BI tools.) If BI tools do not have access to organized data, then it cannot perform its functions.

- Interpretation of Result: It is not enough to process data and generate information from the data. User of BI tools should be in a position to understand and interpret the result generated by the tools.

- Result Analysis: Analysis of results generated by BI tools is needed in order to gain leverage on BI functions. Good result analysis is needed for strategic decision making, prediction, and forecasting.

- Distribution and Sharing of Information: A good BI tool should facilitate the distribution and sharing of its analytical result. Effective communication of results and information from BI tools is needed within an organization in order to execute BI projects. Thus, an individual cannot effectively execute BI projects without communication, sharing information, and results from BI tools with others.

2.1.3.2 Function of BI Tools

As stated in section 2.1.3.1, BI tools are made up of back-end and front-end tools which perform various functions in-order to enable a business entity to gain a competitive advantage by making strategic decisions which will ensure good returns on its investment. The functions of BI tools have evolved over a period of time and are grouped into five styles of BI [82]. The five styles of BI by [82] are as enumerated below:

- Executive/Enterprise reporting: Executive/Enterprise reporting is targeted at customers and company executives (e.g., Chief Executive Officer (CEO)). It is used in reporting

the operational and financial status of the enterprise, showing the overall performance of the organization. Scorecards, dashboards, and portals are used in Enterprise reporting.

- Cube Analysis: This is an OLAP slice and dice of the customer record. This is basically used by non-computer professionals such as the managers to obtain basic information about the performance of the organization on a particular line of business as it operates with data sets of the database that are considered safe and well protected by the system.

- Adhoc Query and Analysis: This style of business intelligence conducts a full investigative query of the entire database record and all the transactions that occur within the organization. It involves querying and slice-dice of the database to obtain information which was not possible using Cube Analysis or Enterprise reposting. This style is used by information explorers to obtain information about the lowest level of transaction details.

- Data Mining/Statistical Analysis: This style segments customer records with a view of determining reports for effective marketing and customer-relationship management. It is used in a particular pattern in the customer record and also in customer behavior toward the organization's goods and services. Data Mining and Statistical Analysis tools include classification, clustering and regression analysis tools.

- Report Delivery and Alerting: This style of BI generates all the types of reports and alerts which include transaction, analytical, and executive reports. This style also gen-

erates reports and sends alerts when any event is triggered or fired in the database. It has the flexibility of allowing reports to be scheduled for distribution to all stakeholders that use the BI tool.

2.1.3.3 Benefits of BI

In-order to derive the maximum satisfaction from BI tools deployment, the BI infrastructure has to function effectively and efficiently. Also, the BI tools and infrastructure are used in enhancing the performance of the business entity [90] by enabling managers to make strategic business decisions and providing easy access to information to all users of BI tools. BI tools help a business entity to gain leverage on the information from the legacy data, operations data, external data, and the integration of these data across the organization. Rosella [122] summarized the benefits of deploying BI infrastructure as follows:

- Simplicity and Ease of Implementation: BI is simple, fast and easy to use tool due to the implementation of its dashboards and analytic reports; changing requirements can be addressed more quickly. It also has low maintenance efforts, high performance, and low total cost.

- Reports: BI is used in producing Operational, Analytical and Executive/Enterprise reports. Thus, all levels of users of BI tools are empowered to use reports suited for their purpose.

- Response Time: Any change in the business requirements is addressed in a timely manner without affecting the normal business operations if BI is implemented.

- Maintenance cost: It costs less to maintain BI infrastructure data because the data is integrated and consolidated in a data warehouse; thus, it is easy to add or delete data.

- High Performance: BI provides one platform access to all the data needed for running a business entity. It provides statistical, analytical, and operational tools that are used by all levels of users to access data. BI tools are easy to use and produce results in a timely manner, thereby enabling strategic decision-making.

2.1.4 Application of BI to Customer Churn Analysis

Customer Churn is the term used to describe the movement of customers from one provider to another provider in the same line of business [8]. The loss of customers in a business entity leads to low return on investment and low profitability. Therefore, customers have to be properly managed in order to retain them and make them loyal to the organization. When a customer churns, there is always the tendency for a business entity to swiftly acquire new customers. But statistics and research have shown that it costs more to acquire new customers than to manage and retain them. Customer Churn Management involves the use of business intelligence tools, integrated with sales and marking skills. Hence, customer churn management is part of customer-relationship management that involves the use of BI

infrastructure to ensure customer loyalty, good return on investments (including maximum profitability) [50, 147]. Customer Churn Management can be done in two ways:

1. By predicting those that are likely to move to a business competitor.

2. By adopting strategies to retain existing customers and make them loyal.

BI techniques are used in predicting and forecasting churners, while marketing and sales techniques are used in making a customer loyal to a business entity. BI tools apply datamining techniques to determine customers that will churn given a database of customer records. A business entity that employs BI tools in its operation has a competitive advantage over others since it uses BI tools to make strategic business decisions such as good business performance and high yield on investments.

2.1.4.1 Types of Business Intelligence Software

The purpose of this section is to review the BI solutions available in the market and in literature. Our analysis shows that most BI softwares in market are not suited for churn prediction, especially when the data are skewed, large, and distinct. Our proposed solution addresses the anomaly in predicting customers who will churn in a business environment. Some of the BI solutions we review are as stated below:

- Microstrategy9: This is a BI solution by Microstrategy [81]. It has all the functionalities of BI tools as specified in 2.1.3.2. In addition, it has performance utilities that

tend to identify each individual performance in a organization when compared to the overall corporate performance of an organization.

- Cognos8 Business Intelligence: This BI software is developed by [54]. It has all the functionalities of BI tools as specified in 2.1.3.2. This BI is unique when compared with other BI solutions in the market as mobile services. The mobile services allow users to receive reports and interact with the Cognos 8 software through their hand held devices.

- SAP BusinessObjects XI: This software is developed by SAP [114]. It has all the functions of BI tools as specified in 2.1.3.2. This solution provides a common BI platform to securely access confidential business information.

- SQL Server2008 Business Intelligence: SQL Server2008 Business Intelligence tool is developed by Microsoft ® [80]. This solution is built on the SQL back-end engine to provide scalable BI platform by integrating large volumes of data items into data warehouse. It has all the functions of BI tools as specified in section 2.1.3.2.

- Oracle Business Intelligence Suite Enterprize Edition Plus (Oracle BI EE Plus) : This solution is developed by Oracle. Like SQL Server2008. it is built on the oracle database to deliver a scalable BI platform that is robust and integrates data items to a large data warehouse. It has all the functions of BI tools as specified in section 2.1.3.2. It also has mobile service functions like Cognos 8 Business Intelligence which delivers reports to

hand-held mobile devices and allows users of the device to interact with Oracle BI EE Plus solution [91].

2.1.5 Data Warehousing

A data warehouse can be described as a collection and integration of all types of data in various formats (text, records from data bases, multimedia data, and image data), including legacy data, operational data, external data, and databases that will be used in the decision support system of a business entity [56]. It is like an information data managing system that involves all the sectors of an organization. DSS as described and defined in section 2.1.3.1 supports decision-making in a business entity and ensures good enterprise management, business process improvement, good returns on investments, and profit maximization. Data Warehouse has been defined in various ways by many authors but the most dominant one is the definition by W. H. Inmon [56] who is regarded as the father of Data Warehouse and one of the first authors on Data Warehouse. Inmon [56] defined data warehouse as "a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decision making process"[56]. This definition by W. H. Inmon is explained as follows:

- Subject-oriented: Data items in a warehouse are stored and organized based on a common subject areas so that all the items are related e.g customer, personnel, order records, and others [34, 56].

- Time-variant: This implies that enhancements and updates to the data ware house are recorded; thus, a report could be produced showing the changes and the inclusion of new data item.

- Non-volatile: The data items stored in the data warehouse are supposed to be static; if the transactions have been committed, they could not be rolled back. Thus, data in the data warehouse could be deleted or removed.

- Integrated: Data in the data warehouse is an integration in various formats, from various dapartments/units of an organization, including operational, legacy and external data items. Despite the source of data, data in data warehouse should be consistent and current which enables data access in real time. Also, duplication of data items should be avoided.

But Kimbal [65] defines Data Warehouse as "a copy of transaction data specifically structured for query and analysis" [65]. Kimbal's approach of building a data warehouse is that of "bottom-up" methodology in which data warehouse is made up of data marts (subsets of data warehouse) while Inmon methodology of building and designing data warehouse is that of "top-down" approached [56]. In the top-down approach, data warehouse includes data from various sources (legacy, operational, and, external) and various formats. The subsets of the data warehouse (data marts) are created after the complete data warehouse is built. But in the bottom-up methodology, conglomerates of data marts make up a data warehouse.

The sources of data items that make up Data Warehouse are so diverse (enterprise-wide frame work) that they have to be processed, organized, summarized and analyzed so that patterns of behavior can be identified [15]. The identification of this pattern in the data warehouse is used business decision making, thus data warehouse is part of the DSS already discussed. The process of building a data warehouse is as stated below:

- Business Requirement Identification: The business requirements of the organization that need a data warehouse have to be identified so that data models can be developed based on the requirements. This will determine the type of data warehouse that is needed.

- Data Extraction: Data items are extracted from many heterogeneous sources and systems then integrated into one common source.

- Data Cleansing: After extracting data item from many sources and formats, the data have to be cleaned to ensure that good data quality is used in the data warehouse. Data cleaning ensures that data redundancy, duplicate data, and noise are eliminated.

- Data Loading. Data that have been cleansed are then loaded into a common source. Thus pattern detection and analytical reporting can be done; these tools will be used in decision making.

Data warehouse architecture is basically made of the meta-data, which contain the data directory. The data source layer contains the raw data that have been summarized, analyzed

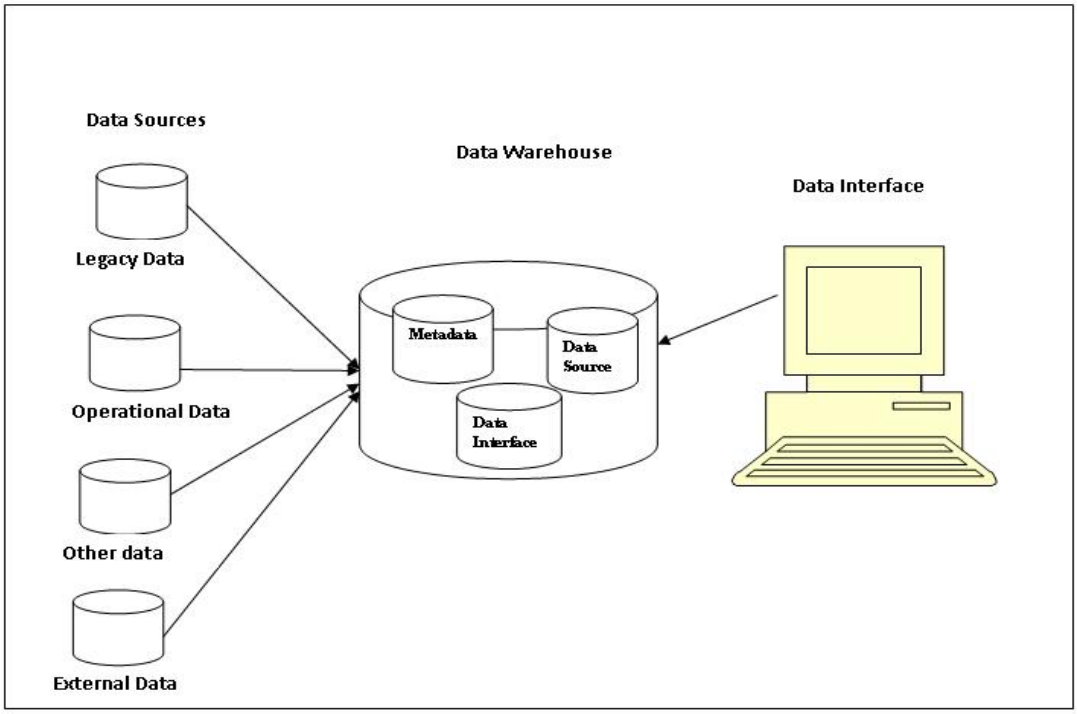

Figure 2.3: Data Warehouse Architecture [18]

and processed. The architecture also contains the interface layer which is the layer through which the data items in the data warehouse are accessed to produce reports and charts which enable its users to make decisions.

Figure (2.3) shows the architecture of a data warehouse.

2.1.5.1 Data Marts

Data Marts are subsets or subtypes of a data warehouse which deal mostly with the data items of a particular product or unit of an organization. It is not as large as a data warehouse. Inmon defines data mart as "a subset of a data warehouse that has been customized to fit the needs

of a department or business process" [56]. Kimball [64] defines data warehouse as "...The data warehouse is nothing more than the union of all the data marts..." [56]. Unlike the data warehouse, the design of a data mart is to meet the needs of the user instead of the data items that are available within an organization.

Data Mart is designed with the aim of addressing user requirements for a particular business process or department. Data Marts are used in DSS because of their ease of access to data items, ease of creation, and good response time with data access. A Data Mart may be dependent or independent of a data warehouse [55]. A data mart that is dependent on a data warehouse derives its data source from the data warehouse, whereas an independent data warehouse may be stand-alone data source.

2.1.6 Data Mining

Data mining is a technique used in searching, sorting, and processing large amount of data in order to discover and extract useful patterns and information [130, 31]. It is a combination of the transactional data analysis method of data processing and analysis and advanced sophisticated algorithms for data sorting and processing [130].

Pattern discovery and its extraction are used by data mining algorithms to identify the relationships among data items. Pattern extraction is used in identifying hidden information from a data source which may be a database, data warehouse or a data mart. Pattern is defined as: "A statement S in L that describes relationships among a subset of facts Fs of a given set

of facts F, with some certainty C, such that S is simpler than the enumeration of all facts in Fs" [31], [117].

Organizations generate several bytes (gigabytes) of data on daily bases as a result of the use of modern computer-based technologies. In fact we are experiencing an information overload because of large volumes of data that are being generated. Thus, manual analysis of the large volumes of data becomes a difficult task. A computer-based technology that can easily search, sort, and extract hidden information is needed; this gives rise to data mining tasks. Data mining can then be described as a computer-based technology that applies different types of sophisticated data analysis algorithms or tools to analyze large volumes of data in order to discover knowledge and predict future events [65]. Thus data mining can be viewed as knowledge discovery and prediction/forecast processes. The tools or algorithms include but not limited to the following: decision tree; neural networks; regression algorithms; and mathematical, statistical, and clustering algorithms. Prediction is the use of known attributes from a data set to determine or forecast the unknown attributes for the given data set while knowledge discovery from a data source is mapping and description of large volumes of data into simple data sets or reports that can be easily be interpreted [31]. Thus, data mining is a subset or step in knowledge discovery and management. Knowledge discovery uses Data Mining process as part of the overall process of the Knowledge Management system which has been discussed in section 2.1.2.

2.1.6.1 Data Mining methods

Data mining tasks is generally grouped into four methods or tasks [31]:

- Classification Algorithms: This is a supervised learning procedure which maps or classifies data items in a data set using a predefined class label. Classification Algorithms include; decision tree, neural networks, nearest neighbor, and Naive Bayes classifier. Classification algorithms are discussed in detail in chapter 4.

- Clustering Algorithms: This algorithm is used to identify or discover and then describe a natural pattern of relationship among data items in a data set. It is an unsupervised learning, unlike classification algorithm as there is no predefined class label. Clustering algorithms include K-means, agglomerative hierarchical, and DBSCAN algorithms. Clustering algorithms are discussed in detail in chapter 3.

- Association Rules: This is used in identifying the hidden relationship among attributes or fields in a large data set [130]. The relationships determine how the data items in a large data set depend on each other. The identified relationship then forms an association rules of "*frequent item sets"* [93]. The data items that are frequent items indicate a good relationship or association among the items's sets; such relationship is leverage upon for forecasting/prediction and in making sales/markeing strategies. The Association rule uses support and confidence measures to determine the strength of data items in a data set [130]. The support measure is used in pruning or eliminating

some of the data items from the data set which may not be profitable to a business entity, while the confidence measure determines the realibility of the rule.

One of the Classical Algorithms of association rules in data mining is the Apriori Principle. It states, *"If an itemset is frequent, then all of its subsets must Also, be frequent*" [130]. Apriori Algorithm uses a breath-first greedy algorithm and tree data structure to identify a subset of the item-set that has the least amount of support. It scans a data set, a couple of times and constructs a candidate set C_k of the frequent B itemset where B is the number of scans of the data set [93, 1]. The candidate set C_k is constructed from previous scan/pass of the itemset as the Apriori principle states, *"If an itemset is frequent, then all of its subsets must Also, be frequent*" [130]. The support and confidence measures of the constructed candidate sets are calculated to determine which one has minimum support. The candidates items with support less than the minimum are eliminated or pruned. The algorithm terminates when there is no frequent data item left in the itemset.

- Regression Algorithms: Regression analysis is used in modeling the relationship between dependent attributes and independent attributes of data items in a data set with the least possible error. Fayyad et al stated, "It maps a data item to a real-valued prediction variable" [31]. It is used in estimating or predicting the value of the dependent variable when the independent variables are varied or kept constant. There are many forms of regression algorithms such as linear, multiple, and logistic algorithms. In a

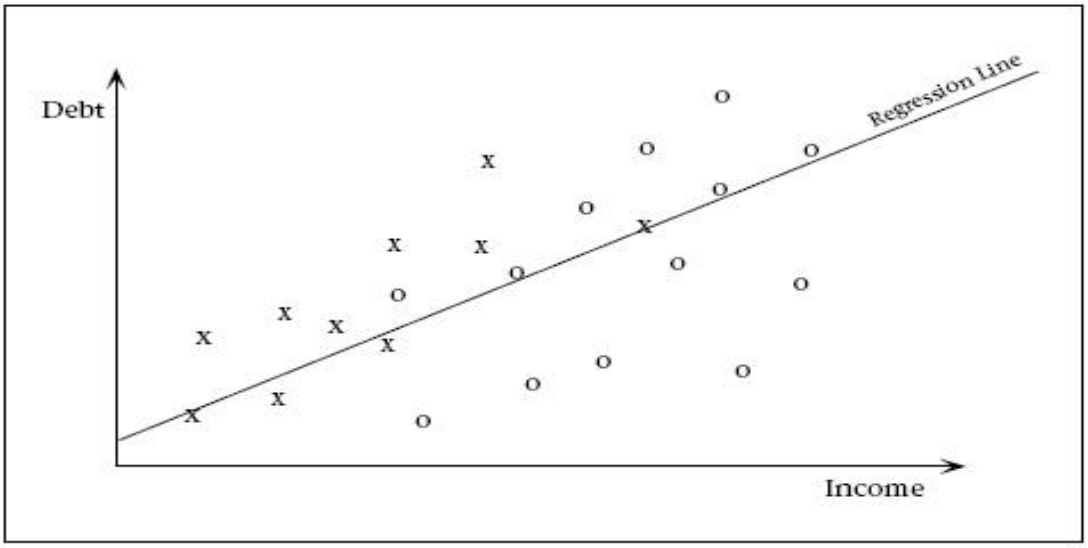

Figure 2.4: Regression for Debt-Income Data set [31]

linear regression, a line of best fit is drawn between the dependent and independent attributes which give a straight line function (equation of a straight line). Figure (2.4) shows a simple regression line between debt and income of a loan management system.

Figure (2.4) shows that there is a relationship or correlation between debt and income in terms of a loan-repayment plan, even though it is a weak one because the line is not well fitted. Logistic regression is used in predicting a discrete outcome of the dependent variable when the independent variable may be categorical or continuous [89].

Chapter 3

Clustering Algorithm

The main concern in the clustering process is to identify the natural organization and patterns in the data by discovering similarities and differences and deriving useful conclusions [138]. Handi et al. stated that clustering analysis can be depicted as a three phase process [45] as shown in Figure (3.1). The first (data pre-processing) phase performs data transformation to include feature selection, data normalization, removal of noise and outliers, and the selection of distance function. The second phase is the cluster analysis; it involves the selection of the clustering algorithm and its parameters and the application of the selected algorithm. The final phase in the cluster analysis is the verification (validation) which involves selection of validation technique and its evaluation. In this phase the quality of the partitioning and the clustered data are evaluated [45]. If the quality of cluster is below expectation, then the whole process is repeated until the desired quality is obtained.

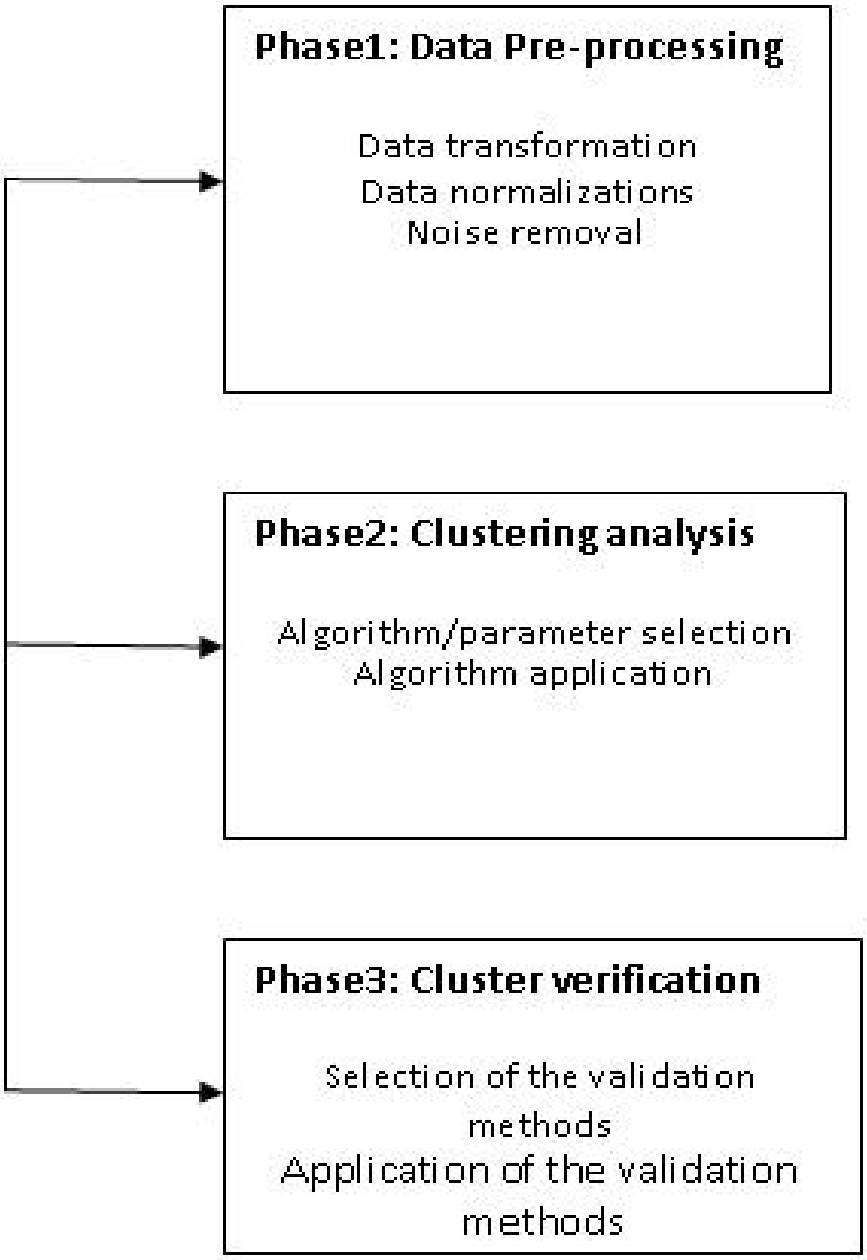

Figure 3.1: The three steps of clustering

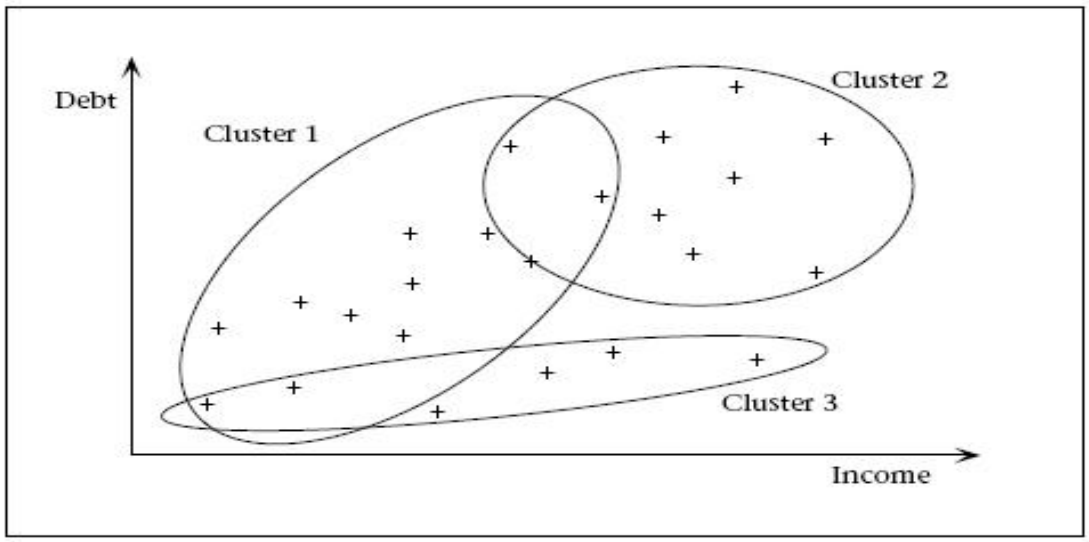

Figure 3.2: Clustering of Loan Data set into three clusters[31]

Figure 3.2 shows the clustering of loan data set into three clusters (cluster1, cluster2, and cluster3). The overlapping of the clustering structure (cluster1 and cluster3) shows that a data item can belong to two clusters.

3.1 Cluster Validity

Since clustering is an unsupervised classification technique with no predefined classes or classification examples, the final partitions of a data set require some sort of evaluation in most applications. Therefore, cluster evaluation, or cluster validation as it is more tradition-

ally called, has become an essential part of any cluster analysis. The major problem of cluster algorithms is to identify the optimal number of clusters for a given data set [42]. In most experimental analysis, the evaluation of cluster algorithms in Euclidean space is done visually when the data dimension is less than three [42]. But when high dimensional data sets are involved, visual verification of the cluster analysis becomes difficult, hence the need for cluster validity measures (indices) that can validate data sets of multiple dimension. In validating a clustering algorithm, the clustering indices are applied to both the algorithm development and the output or result of the clustering algorithm. The application of cluster indices to algorithm development is used in determining the effectiveness of a particular clustering algorithm and the type of data set being used by the algorithm. When a clustering index is applied to the result of a clustering algorithm, the objective is to obtain a detailed analysis of the clustering algorithm. Result verification shows that the optimal cluster set is obtained by verifying the structure of the partitioned data [42, 43]. The following section discusses different kinds of cluster validity measures.

3.2 Cluster Validity Measures

A wide variety of clustering validity methods have been proposed in literature. In general, these methods are based on three approaches [132]: internal, external, and relative criteria. External criteria imply that the clustering solution is matched to prior information, i.e., external information that is not contained in the data set. On the other hand, the quality measure

in internal criteria is exclusively based on the data. Finally, in the relative criteria the evaluation of a clustering structure is done by comparing it to other clustering schemes, using the same algorithm but with different parameter values. The three criteria are investigated and summarized below.

3.2.1 Internal Criteria Measures

Internal criteria measures are cluster validity measures which evaluate the clustering result of an algorithm by using only quantities and features inherent in the data set. The cluster validity measure can be done in two different ways: using the cohesion and separation measures or using the similarity (distance) matrix.

3.2.1.1 Cohesion and Separation Measures

Cluster Cohesion is used in measuring the closeness or proximity of the data objects in a cluster while cluster separation is used in measuring how one cluster is separated from other clusters. Cluster cohesion and separation measurements are shown in (3.1):

$$cohesion(C_i) = \sum_{x \in C_i, y \in C_i} proximity(x, y) \tag{3.1}$$

$$separation(C_i, C_j) = \sum_{x \in C_i, y \in C_j} proximity(x, y) \tag{3.2}$$

C_i and C_j are set of cluster objects.

The cohesion and separation techniques are stated in the following sections:

1. **The Davies-Bouldin Index (DB)**: The DB [25] index measures the cohesion (compactness) of the clusters where small values correspond to compact clusters. DB index is defined in (3.3):

$$DB = \frac{1}{K} \sum_{i=1}^{k} \max_{j \neq i} \left[\frac{diam(C_i) + diam(C_j)}{d(C_i, C_j)} \right] \tag{3.3}$$

   where K is the number of clusters, C_i and C_j are two clusters, $diam(C_i)$ is the average distance of all objects in cluster C_i to the center (cluster diameter) and $d(C_i, C_j)$ is the distance between the two clusters. This distance is small if clusters i and j are well separated and each of the clusters is compact. The DB index takes value within the range of $[0, 1]$.

2. **The Dunn Index(DI)**: Dunn (1974) has a very popular cluster validity index used to identify clusters with high cohesion and separation, known as the Dunn index [29].

It is used in the identification of clusters that are well defined. The index of Dunn is defined in (5.2):

$$D(C) = \min_{i=1,\cdots,n} \left\{ \min_{j=i+1,\cdots,n} \left\{ \frac{d(C_i, C_j)}{\max\limits_{h=1,\cdots,n} \{dm(C_h)\}} \right\} \right\} \tag{3.4}$$

where C_i and C_j are the closest clusters according to the distance d, and C_h is the cluster with the largest diameter. Since D(C) only depends on few clusters and the distance between them, and both $d(C_i, C_j)$ and dm (C_h) are sensitive to outliers, the Dunn's index is not always reliable. The distance between clusters C_i and C_j is $d(C_i, C_j)$ which is defined as:

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$$

The total number of clusters is n and $dm(C_h)$ is the inter-cluster distance(diameter) of C_h, which is defined as:

$$dm(C_h) = \max_{x,y \in C_h} d(x, y)$$

The Dunn index takes value within the range of $[0, 1]$.

3. **Silhouette Index (SI)**: Like the Dunn index, Silhouette index [109] combines cohesion and separation. The Silhouette index $(s(v_i))$ for a single object v_i in a cluster C_j is computed as shown in (3.5):

$$s(v_i) = \frac{d(v_i,C_h)-d(v_i,C_j)}{max(d(v_i,C_j),d(v_i,C_h))} \tag{3.5}$$

C_i and C_j are sets of cluster objects that are close to each other. C_h is the cluster with the largest diameter. Given that $d(v_i, C_j)$ is the average distance between v_i and all the other objects in C_j, and C_h are the closest clusters to the object v_i that does not contain v_i. Note that the value of $s(v_i)$ can vary between -1 and 1 (Note: larger values are better.). A negative value is undesirable since it means that the average distance to points in the cluster is larger than the minimum average distance to points in another cluster. The silhouette index (S_j) for the whole C_j cluster is defined as

$$S_j = \frac{\sum_{i=1}^{N_j} s(v_i)}{N_i} \tag{3.6}$$

where N_i is the number of objects in the cluster C_i , N_j is the number of objects in the cluster C_j. The global silhouette (GS) for all the clusters is calculated as:

$$GS = \frac{\sum_{j=1}^{k} S_j}{K} \tag{3.7}$$

The value Silhouette index is within the range of $[-1, 1]$.

4. **C-Index** [49]: Let D be the sum of all values within-cluster dissimilarities, D_{min} and D_{max} be the minimum and maximum sums of the values within cluster dissimilarities in the baseline distribution. Cluster similarity is a numerical measure of the degree to which two cluster objects look alike, while cluster dissimilarity is a numerical measure of the degree in which two clusters are different. The C-index is defined by:

$$C = \frac{D - D_{min}}{D_{max} - D_{min}} \tag{3.8}$$

It is obvious that the numerator in the above formula will be small for pairs of objects with a small distance. Hence, a small value of C-index indicates a good clustering and gives a measure of the closeness of the objects to the clusters.

3.2.1.2 Similarity Matrix

1. **Correlation Method**: Given the similarity matrix of a data set and the cluster labels from a cluster analysis on the data set, clustering can be evaluated by looking at the correlation between the similarity matrix and an ideal version of the similarity matrix based on the cluster labels. The ideal similarity matrix is constructed by creating a matrix that has one row and one column for each pair of points with a value of 1 if it belongs to the same cluster; otherwise, it has 0 value if the pair of points belong to different individual cluster [24].

3.2.2 External Criteria Measures

The following techniques are based on external criteria in the measure of validity. Just like the internal criteria, the external criteria measurement can be done in two ways. The first one is that the clustering structure C is evaluated by comparing it with an independent partition of the data set P built according to our intuition of clustering data set. The second way is to compare the proximity matrix with the partition P.

3.2.2.1 Cohesion and Separation Measures

Filzmoser et al. [33] used the following procedure of Rand Index, Jaccard Coefficient, Folkes and Mallows index, and Hubert's Γ Statistic external measure techniques in validating data sets. If $C = C_1 \dots C_n$ is a clustering structure of a data set, X and $P = P_1 \dots P_n$ is the

Table 3.1: Data Partition Set

	A	B
A	a_1	a_2
B	a_3	a_4

defined partition of the data set. At different points in the data partition set, the following cases are possible as shown in Table 3.1 [33]:

- **AA**: Both points belong to C and P.
- **AB**: Both points belong to C and different groups of P.
- **BA**: Both points belong to P and different cluster of C.
- **BB**: Both points belong to different clusters of C and to different groups of P.

Table 3.1 shows that data points a_1, a_2, a_3 and a_4 are numbers in AA, AB, BS, and BB respectively. $Z = a_1 + a_2 + a_3 + a_4$. The degree of similarity between C and P is defined by the following validity methods:

1. **Rand Index (RI)**: [104] This measure considers the pair of a data points in each cluster set.

$$R = \frac{a_1+a_4}{Z} \tag{3.9}$$

2. **Jaccard Coefficient(JC)** [57]: This coefficient measures the proportion of pairs of points that belong to the same cluster but appears in both partitions.

$$J = \frac{a_1}{a_1+a_2+a_3} \tag{3.10}$$

Both Rand and Jaccard coefficients take indices between 0 and 1.

3. **Folkes and Mallows index (FM)**: [41] This index measures the geometric mean for that portion of the pairs of points for the same cluster which exist in all the partitions. The equation for the index is given below :

$$\frac{a_1}{\sqrt{(a_1+a_2)(a_1+a_3)}} \tag{3.11}$$

4. **Hubert's Γ Statistic [132]**

$$\Gamma = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} x_{ij}y_{ij} \tag{3.12}$$

M = n(n-1)/2, n is the number of points in the data sets, x_{ij} and y_{ij} are the elements of the matrices P and C, respectively, that are to be compared. In all the above indices a high value indicates a high similarity between C and P.

5. **Partition Coefficient** [9]: This measure is used to evaluate the performance of a classification model. It states that if the data classification is good, then data can be assigned to clusters with membership degrees between 0 and 1 [33]. The partition coefficient (PC(U)) is defined in (3.13):

$$PC(U) = -\frac{1}{n}\sum_{i=1}^{n_c}\sum_{j=1}^{n} U_{ij}^2 \tag{3.13}$$

Here n_c is the number of clusters and n is the total number of data points and u_{ij} $(i = 1, 2, \ldots n_c;\ j = 1, 2, \ldots n)$ is the membership of data point j in cluster i. The closer this value is to 1 the better the data is classified. The range of PC values is within the interval $[-1, -\frac{1}{n_c}]$

6. **Partition Entropy** [9]: This behaves in the same manner as the Partition coefficient but the partition entropy is with the range $[0, \ln(n_c)]$. The partition entropy is given by the following equation:

$$PE(U) = -\frac{1}{n}\sum_{i=1}^{n_c}\sum_{j=1}^{n} U_{ij}\ln(U_{ij}) \tag{3.14}$$

7. **Purity**: Another measure of classification performance is purity. It is a measure that shows to which limit a particular cluster contains objects that are of the same class. The purity measure is also known as the maximum precision over all classes in the cluster [24] . The purity of a cluster i is given by:

   $pi = \max(p_{ij})$

   The overall purity of a clustering is given in (3.15):

$$Purity = \frac{m_i}{n_c} \sum_{i=1}^{k} pi \tag{3.15}$$

The number of cluster objects in cluster i is m_i is while n_c is the total number of cluster objects.

8. **Compactness and Separation**: This measure is published by Xie et al. (2000) [145]. In this measure there is a comparison between the distance of the data to the clusters and the distance between the clusters. The formula for the measure is given in (3.16).

$$S(U, X) = \frac{\sum_{i=1}^{n_c} \sum_{j=1}^{n} U_{ij}^2 . d^2(x_j, v_i)}{n. \min\{d^2(v_i, v_j) | i, j \epsilon \{1, \ldots n_c\}, i \neq j\}} \tag{3.16}$$

The center of cluster C_i is v_i and v_j is the center of C_j. The number of clusters is n_c and n is the total number of data points and U_{ij} $(i = 1, 2, \ldots n_c;\ j = 1, 2, \ldots n)$ is the membership of data point j in cluster i. The density of the clusters is U which is defined in (3.17).

$$density(U) = \sum_{i=1}^{n_{ij}} f(x_i, U) \tag{3.17}$$

The number of tuples that belong to the cluster (number of data points around U) C_i and C_j is n_{ij}. The function f(x,U) is defined as stated below:

The value of f(x,U) is 0 if d(x,v) is greater than the average standard deviation, otherwise 0.

In (3.16) the numerator of this equation represents the data within a cluster (homogeneity of data), and its value is normally small while the denominator evaluates data from different clusters (heterogeneity of data). The smaller the value of (3.16), the better the classification/clustering.

3.2.2.2 Similarity Matrix

The same procedure described above that creates an ideal similarity matrix and compares it to the actual one is used. The Γ Statistic index could be used after that as an indication of the two matrices's similarity [33].

3.2.3 Relative Criteria Measures

The basic idea in relative criteria is the evaluation of a clustering structure or cluster by comparing it to other clustering structures or clusters, resulting from the same algorithm but with different parameter values. Thus, relative measures are not actually a separate type of cluster evaluation measures. Internal or external measures (e.g., partition entropy) are used in this type of evaluation.

3.3 Validity Measures Statistics

In an attempt to find which of the previous measures are used more often in clustering experiments, thirty-two papers that contain clustering validity experiments/study cases have been reviewed and the validity measures used are recorded for each paper. The results of the survey are summarized in the tables below where papers are represented by the last name of the first author. Table 3.2 shows the survey of all the papers reviewed. Table 6.1 shows the frequency usage of each of the measure. Table 6.2 shows the summary of the internal measure usage while Table 3.5 shows the external measure usage. Figures 3.3 and 3.4 show the frequency usage of the internal and external measures and all the validity measures respectively.

Table 3.2: Literature review of cluster validity measures

Paper	**Validity Measures**
Chen et al. 2006 [19]	Jaccard coefficient
Bolshakova et al. 2006b [10]	C-Index, Silhouette, Dunn's, Davis-Bouldin
Chou, Su and Lai, 2003 [20]	DI DB PC CE S
Filzmoser et al, 2003 [33]	PC, PE, Separation-Compactness
Murata, 2003 [86]	Jaccard coefficient
Stein et al. 2003 [127]	Dunn's, Davis-Bouldin
Kuncheva and Vetrov, 2006 [67]	Rand
Watanabe et al. 2005 [142]	Jaccard coefficient
Zhao et al. 2006 [150]	Hubert's Γ Statistic
Azuaje, 2002 [4]	Dunn's
Xie et al. 2000 [145]	Compactness-Separation
Kasturi et al. 2003 [62]	Davis-Bouldin, Rand Index
Gepas, 2005 [38]	Silhouette
Zimmermann et al. 2004 [151]	Rand index
Yang et al. 2005 [148]	C-Index
Loganantharaj et al. 2006 [75]	Dunn's
Huang et al. 2001 [48]	C-Index
Sadesky (n.d) [112]	C-index
Liu et al. 2005 [74]	Dunn's, Silhouette, C-Index, DB
Van et al. 2002 [136]	C-Index
Johansson and Lindberg, 2000 [59]	C-Index
Barbaranelli,2002 [5]	C-Index
Dunn,1974 [29]	Dunn's
Sudhakar and Rajagopalan, 2004 [128]	Dunn's and Silhouette
Chunmei et al. 2006 [21]	Dunn's, Silhouette, Davis-Bouldin
Bolshakova et al. 2003b [12]	C-Index, Silhouette, Dunn's, Davis-Bouldin
Bolshakova et al. 2003a [11]	C-Index, Silhouette, Dunn's, Davis-Bouldin
Rousseuw,1987 [109]	Silhouette
Davies and Boudlin,1979 [25]	Dunn's, Davis-Bouldin
Bolshakova et al. 2005 [13]	C-Index, Silhouette, Dunn's, Davis-Bouldin
Yeung et al. 2001 [149]	Dunn's
Wang et al. 2008 [140]	Cluster Validity, method for support vector cluster
Bolshakova et al. 2006a [14]	Silhouette, Dunn's, Davis-Bouldin
Leg*á*ny et al. 2006 [69]	SD Validity Index, Dunn's, Davis-Bouldin

Table 3.3: Frequency use of each measure

Measure	Frequency Use
Internal	77.6 %
External	22.4 %
Jaccard coefficient	5.2 %
Silhouette	15.5 %
Partition Coefficient	1.7 %
Partition Entropy	1.7 %
Davis-Bouldin	18.97 %
Dunn's	24.1 %
Rand Index	5.1 %
C-Index	18.97 %
Hubert's Γ Statistic	5.1 %
Compactness-Separation	6.4 %

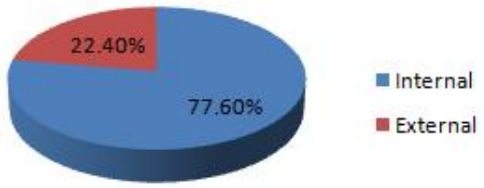

Figure 3.3: Internal and External Measure Frequency Usage

Table 3.4: Summary of Internal Measure Statistics

Paper	**Validity Measures**
Bolshakova et al. 2006a [14]	C-Index, Silhouette Dunn's, Davis-Bouldin
Chou, Su and Lai, 2003 [20]	Dunn's,Davis-Bouldin
Stein et al. 2003 [127]	Dunn's, Davis-Bouldin
Azuaje, 2002 [4]	Dunn's
Kasturi et al. 2003 [62]	Davis-Bouldin
Gepas, 2005 [38]	Silhouette
Yang et al. 2005 [148]	C-Index
Loganantharaj et al. 2006 [75]	Dunn's
Huang et al. 2001 [48]	C-Index
Sadesky (n.d) [112]	C-index
Liu et al. 2005 [74]	Dunn's, Silhouette C-Index, DB
Van et al. 2002 [136]	C-Index
Johansson and Lindberg, 2000 [59]	C-Index
Barbaranelli, 2002 [5]	C-Index J.C
Dunn, 1974 [29]	Dunn's Sudhakar
Sudhakar and Rajagopalan, 2004 [128]	Dunn's and Silhouette
Chunmei et al. 2006 [21]	Dunn's ,Silhouette Davis-Bouldin
Bolshakova et al. 2003b [12]	C-Index, Silhouette Dunn's, Davis-Bouldin
Bolshakova et al. 2003a [11]	C-Index, Silhouette Dunn's,Davis-Bouldin
Rousseuw, 1987 [109]	Silhouette, J.L Davies and D.W
Davies and Boudlin, 1979 [25]	Dunn's, Davis-Bouldin
Bolshakova et al. 2006a [14]	C-Index, Silhouette Dunn's, Davis-Bouldin
Yeung et al. 2001 [149]	Dunn's
Bolshakova, Anton and Pdraig, 2006b [10]	Silhouette, Dunn's Davis-Bouldin

Table 3.5: Summary of External Measure Statistics

Paper	Validity Measures Used
Chen et al. 2006 [19]	Jaccard coefficient
Chou, Su and Lai, 2003	PC, PE Separation-Compactness
Filzmoser et al. 2003 [33]	PC, PE Separation-Compactness
Murata, 2003 [86]	Jaccard coefficient
Kuncheva and Vetrov, 2006 [67]	Rand
Watanabe et al. 2005 [142]	Jaccard coefficient
Zhao et al. 2006 [150]	Hubert's Γ Statistic
Xie etal, 2000 [145]	Compactness-Separation
Kasturi et al. 2003 [62]	Rand Index
Zimmermann et al. 2004 [151]	Rand index

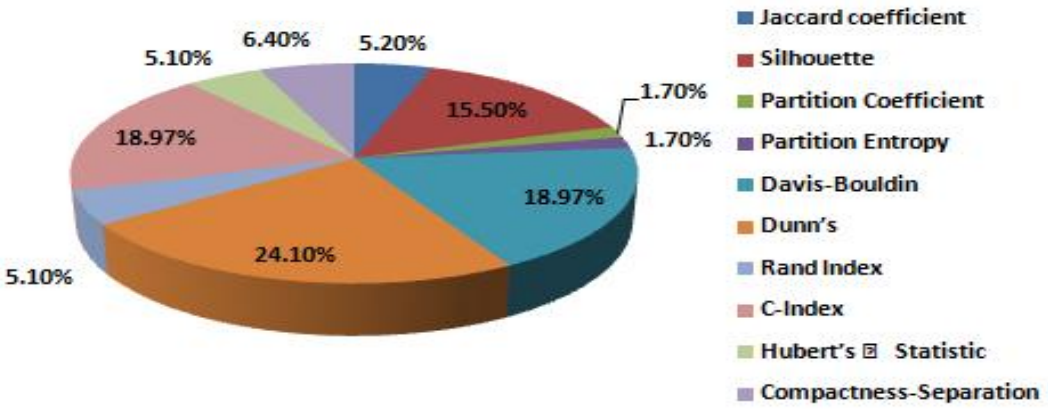

Figure 3.4: Validity Measures Frequency Usage

3.4 Experimental Analysis

We developed a system that validates the different kinds of cluster validity measures. The system takes in information about the data to be clustered (original data) and results produced by a clustering algorithm (clusters). It then measures the validity of the clustering algorithm based on proximity measures. The measures used are divided into external and internal measures. With internal measures we implemented Dunn, Davies-Bouldin and Silhouette, while for external measures we implemented Rand, Jaccard, Folkes and Mellow, Purity Folkes and entropy. In the validation, we used the following kinds of data sets: original, cluster results, tuple based, distance based and half-distanced based matrix data sets. The data sets are computer generated to conform to the given specification stated in section 6.1. The user is allowed the flexibility of choosing which of the measures (internal or external) and also, to choose the required data set for the validation. The system is developed using a user friendly tool and run in either batch mode or GUI environment. Figure 3.5 shows the interface for the developed cluster validity system.

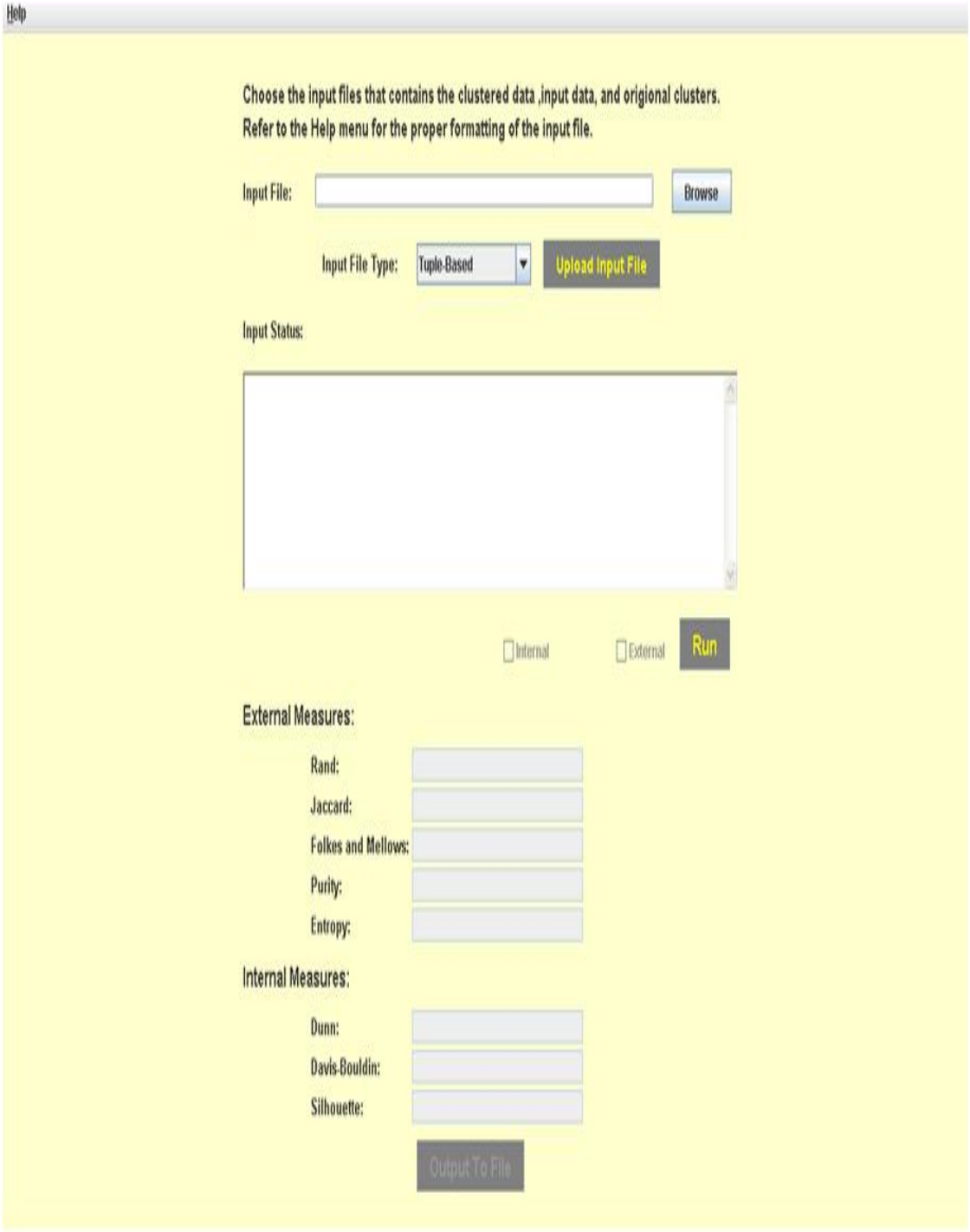

Figure 3.5: Developed Cluster Validity System

3.4.1 Data Set Usage

In developing the system, we used different types of data sets: tuple-based, distance and half distanced matrices, original data sets and cluster result (clusters). All the data sets used are computer generated data sets. The format of the data sets used is stated in the following sections:

3.4.1.1 Tuple-based Data set

- The first line specifies the total number of objects.
- The second line specifies the total number of attributes per object.
- The third line states the type of each attribute (f : floating pt, i: integer, c: character, s: string [strings have to be enclosed by " "]).
- Each line is a record/object.
- Each object will have the attributes specified as in line 3 above.
- Attributes are separated by a blank space.

3.4.1.2 Distanced-based (matrix) Dataset

- The first line specifies the total number of objects.
- Then each line specifies the distance between an object and all others (in N x N matrix).
- If the first line is negative, then only the upper left half matrix is represented.

3.4.1.3 Distanced-based (half matrix) Dataset

- The first states the total number of objects.
- Then for each line, it specifies the distance between an object and all others (N x N matrix).
- Only the lower left half matrix is represented (because of symmetry).

3.4.1.4 Original Data set

- The first line specifies the total number of objects.
- Each line corresponds to the cluster number of that object (represented by integers).

3.4.1.5 Cluster Result(cluster) Data set

- The first line specifies the total number of objects.
- The second line specifies the total number of cluster results.
- Each line corresponds to the cluster number of that object (in whatever number of result).

3.4.2 Experimental Results

The system was subjected to rigorous tests using 50 different data sets, and five different trials were conducted for each data set. Each of the data sets is formatted using the specifications

Table 3.6: Euclidean Distance Measurement

	Test1	Test2	Test3	Test4	Test5	Average	Standard Deviation
Dunns Index	0.092	0.020	0.099	0.090	0.020	0.064	0.041
Davis-Bouldin	14.956	53.790	52.346	45.900	43.986	42.196	15.783
Silhouette index	0.604	0.720	0.691	0.697	0.686	0.680	0.044

Table 3.7: Cluster and Data set Correlation Measurement

	Test1	Test2	Test3	Test4	Test5	Average	Standard Deviation
Rand Index	0.600	0.564	0.607	0.605	0.612	0.598	0.261
Jaccard coefficient	0.429	0.111	0.154	0.409	0.390	0.299	0.153
Folkes & Mallows index	0.612	0.200	0.267	0.513	0.416	0.402	0.170
Purity	0.800	0.545	0.234	0.725	0.452	0.551	0.225

stated in sections 6.1.1, 6.1.2, 6.1.3, 6.1.4, and 6.1.5 respectively. Also, the same number of cluster data sets is used in all the tests. Both internal and external measures were tested. The system was found to be stable and generate consistent results. The data sets are all computer generated. With the internal measure, the Euclidian distance for the half and full matrix is calculated; for the external measure, the correlation between the data sets and the clusters is compared using various measuring techniques as stated in section 4. The test results and analysis are shown in the Tables 3.6 and 3.7. Table 3.6 shows the internal validity measure test while the external validity measure test is shown in Table 3.7.

3.4.3 Test Observation

- Using the internal validity measure, the goal is to evaluate the result of clustering algorithm. This evaluation is done by using features from the original data set like sepa-

ration, compactness, etc., between the clusters which are Euclidean distance measures. The higher the cohesion value of a cluster, the better the cluster. Thus, the more the results of validation tests become smaller (tend to positive zero), the better the clustering validation measure. The average and standard deviation result in Table 3.6 shows that Dunn's Index has the least value from all the five tests, when compared to Davies-Bouldin and Silhoute indices. Thus, Dunn's Index is preferred among all the other methods as it gives more accurate validity measure than other internal validity measures. Thus, our validation experiment shows that Dunn's index is well positioned to identify compact and well separated clusters when compared with other measures and it conforms with Dunn's index objective, which is to maximize intra-cluster distance, while at the same time minimizing the inter-cluster distance. This objective is verified by our experimental analysis.

- Using the external validity measure, the cluster is compared to an independent partition of the data and cluster; then purity and completeness of the clusters are compared. The result of the comparison is shown in Table 3.7. The test scores with higher values (positive 1) are a good test of validity measures; thus, the higher the value of the test results the more accurate the measure. The average and standard deviation test result in Table 3.7 shows that Rand Index has the highest value; thus, it is the preferred measure compared with other external measures. The high value of the test result for the Rand index indicates that the clusters it validates are pure and complete, it also indicates a high level of agreement between the clustering technique and the original

classes. Thus, when compared to other external validation techniques, Rand index shows that clustering results are closer to its original classes which is what a good clustering technique tends to achieve.

3.5 Summary

Traditional cluster validity measures like density and variance are not used in any of the papers we reviewed. The idea of not using density and variance measure can be attributed to the fact that some clusters can be abnormally shaped, e.g., spatial data; thus, using any of the traditional validity measures may not give accurate results [42]. According to the survey statistics, Dunn's Index is the most common validity measure in the 32 papers surveyed. Also, internal measures are more frequently used than external measures. Our experimental analysis also confirms that Dunn's Index is preferred among the internal measures as it has the least value of the average test scores, while Rand Index is preferred among the external measures as it has the highest test score compared to other external measures from the experimental analysis. Thus, one can conclude that Dunn's index is a good cluster validity measure based on the survey statics and the experimental result as it identifies complete and compact clusters in a clustering technique. But the type of validity measures used will depend to a great extent upon the data set type and the clustered data set. In a future study, we will apply our experimental analysis to data of different format and types and then compare the results with the survey analysis of clustering validity measures.

Chapter 4

Classification and Prediction Algorithm

Classification can be described as a supervised learning algorithm in the machine learning process. It assigns class labels to data objects based on prior knowledge of class to which the data records belong. It is a data mining technique that deals with knowledge extraction from database records and prediction of class labels from and unknown data set of records [130]. In classification a given set of data records is divided into training and test data sets. The training data set is used in building the classification model, while the test data record is used in validating the model. The model is then used to classify and predict a new set of data records that is different from both the training and test data sets [36, 37]. Supervised learning algorithm (like classification) is preferred to unsupervised learning algorithm (like clustering) because its prior knowledge of the class labels of data records makes feature/attribute selection easy, leading to good prediction/classification accuracy. Some of the common classification algorithms used in data mining and decision support systems are as follows: neural

networks [73], logistic regression [63], and decision trees [103]. Among these classification algorithms, decision tree algorithms are the most commonly used because they are easy to understand and inexpensive to implement. They provide a modeling technique that is easy for people to comprehend and simplifies the classification process [135]. Most decision tree algorithms can be implemented in both serial and parallel form while others can only be implemented in either serial or parallel form. Parallel implementation of decision tree algorithms is desirable in order to ensure fast generation of results especially with the classification/prediction of large data sets; it also exploits the underlying computer architecture [116]. But serial implementation of decision algorithm is easy to implement and desirable when small-medium data sets are involved. In this paper, we will review the most common decision tree algorithms implemented serially and perform an experiment to compare their classification and prediction accuracy.

4.1 Classification Algorithm

Classification is a data mining technique that uses a set of known data records to classify or predict future or unknown records [36]. A record data set is divided into training and test data sets which are used in the classification algorithm. The record data set is made up of several attributes or fields which describe the data. Attributes can be categorical (unordered) or continuous (ordered). One of the categorical attributes is called the class label as it is used in classifying the record data set; other attributes are called predictor attributes. Categorical

attributes are used as binary classifiers in churn prediction since they are used to predict churners and non-churners of customers' record data sets. The training data set is used in training the model, while the test data set is used in validating the model [37].

The main idea behind classification algorithm is to use the training set of data records to build a model of the class label attributes, using other attributes; then test data is used to validate the model. Applications of classification algorithm include, but are not limited to, customer churn prediction, credit card fraud detection, credit approval ratings, bankruptcy prediction, medical diagnosis, and so on. Classification is a supervised learning algorithm and has the advantage over other unsupervised learning algorithms (e.g., clustering) since there is a prior knowledge of the class labels to which the training data records belong; thus, feature/attribute selection is used by the algorithm for a good classification accuracy.

A number of classification algorithms like neural networks [73], logistic regression [63], decision trees, [103] and so on, are available in the literature. Decision tree seems to be the most commonly used among the classification models as it is an eager learning algorithm and inexpensive to implement. In this study, we will focus on decision tree algorithms.

4.1.1 Classification Accuracy

The accuracy of a classification model is determined by test cost errors and classification errors (training errors). The test cost errors, often considered minimal are costs incurred in obtaining the attribute values, while classification errors are the costs incurred in misclassifying the attributes from test data. In this study, we will focus on misclassification errors

[71]. It is difficult to build a classification algorithm without having some misclassification errors like false positive (FP) and true negative (TN). In customer churn prediction, binary classifiers used in customers' churn predictions are classified as churners and non-churners. Churners are assigned positive (P) class, while non-churners are assigned negative (N) class; most often some of the classification are wrong leading to misclassification errors, such as TN (a positive assignment that is classified as negative) and FP (a negative assignment that is classified as positive). The correct predicted classes are True Positive (TP) (a positive assignment that is actually classified as positive) and False Negative FN (a negative assignment that is actually predicted to be false). The classification matrix is shown in table 4.1:

Table 4.1: Classification Matrix

PREDICTED CLASS			
Actual Class		Class=yes	Class=No
	Class =yes	TP	TN
	Class= No	FP	FN

The relationship between TP, FN, FP, P and N is given below;

$$TP + FN = P \tag{4.1}$$

$$TN + FP = N \tag{4.2}$$

Other measures used to determine the accuracy and quality of a classification algorithms are: Specificity, Sensitivity, Recall, Precision, FPrate, TPrate, Misclassification Error (MER), ROC (Receiver Operating Characteristic) Curve and Lift Curves.

Recall = $\frac{TP}{P}$ = Sensitivity = TPrate, Precision = $\frac{TP}{TP+FP}$

Accuracy =$\frac{TP+TN}{P+N}$, MER = 1- accuracy

Specificity = $\frac{TN}{N}$ = FPrate, Lift = $\frac{Precision}{P/(P+N)}$

Sensitivity: The percentage of positive classifications that are actually classified as positive.

Specificity: The percentage of negative classifications that are actually classified as negative.

ROC Curve: a graph of Sensitivity against 1-Specificity at many cut-points. The area under the curve is a measure which separates positive and negative classification. An increase in the cut-points leads to an increase in Sensitivity while Specificity decreases [108].

Lift: This measure is used in comparing precision to the churn rate. It is used in comparing the percentage of positive prediction to the percentage of the customers data records.

4.2 Decision Tree Algorithm

Decision tree is one of the most popular classification algorithms because it is easy to implement, an eager learning algorithm, and easy to understand. Also, decision tree algorithms do not necessarily require additional information when building the prediction model. It makes use of the already available record in the training data set in building the model [105]. When compared with other classification algorithms the decision tree algorithm is easy to under-

stand because it is based on rules. The rules can be applied to database retrieval language like SQL as the database contains relational tables which are based on certain categories of data [44].

A decision tree is a classification algorithm that recursively partitions data using breadth-first algorithm until each data is pure; this implies that each partition contains records which are of the same class or a leaf node [116]. This is a greedy approach since at each internal node the best split point is chosen. Figure 4.1 shows the decision tree algorithm with the associated training data set. The leaf nodes have a class (Excellent, Fair, Excellent or Fair) associated with it, while the internal nodes are decision nodes which test the attributes: Age, Loan Amount and Income for some specific values. The outcome of the test results in the tree branching; it would be yes or no for each internal node test [105, 125]. The internal nodes are also, called splitting or predictor attribute because it is at these nodes that splitting decisions are made. In order to classify new records using the decision tree algorithm, the tree is transversed recursively using greedy and breadth-first approach, starting from the root node, through the internal nodes, until the leaf nodes are reached; classification of the new records then occurs at the leaf nodes [105].

There are many decision tree algorithms proposed over the last few years which include the following: C45 [103] , CLS [51] , CART [17] , ID3 [101], Random Tree [139], RainForest [37], SLIQ [78], SPRINT [116], etc. CART, ID3, and C45 are based on Hunt's algorithm (The training data set is partitioned recursively using depth-first approach until the data items are grouped in classes called purer group also, the data items are loaded into the memory at

Sid	Loan amount	Age	Income	Credit rating
0	15	35	70	Fair
1	12	28	20	Excellent
2	12	45	80	Fair
3	15	60	45	Excellent
4	25	60	105	Fair
5	15	50	65	Fair

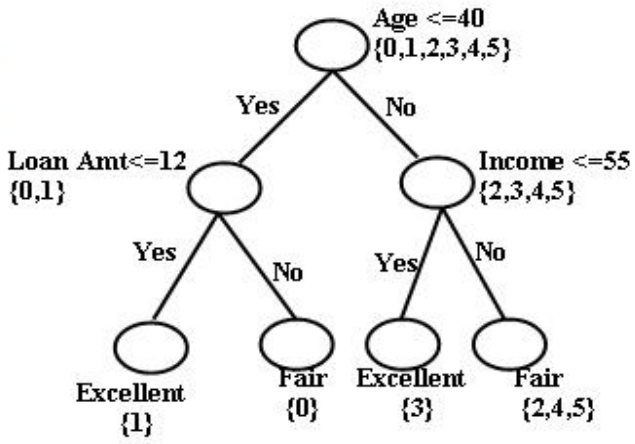

Figure 4.1: Decision Tree Classification Algorithm

the same time). Most of these algorithms classify and predict new data records in two phases: tree building (growth) and tree pruning phases.

4.2.1 Tree Growth

The tree building phase is computationally more difficult and expensive compared with the tree pruning phase since the data set is transversed multiple times while pruning is performed on a fully grown tree. During the tree building stage, the training data is recursively partitioned using breadth-first approach until each partition contains members of the same group or class.

The partitioning of the data depends on the evaluation of the splitting attribute [125]. In partitioning the data, the data type of the splitting attribute is also considered. The splitting point is of the form $A < z$ (A is numerical data) and $A \in z$ (A is categorical data), where z is a member of A data set [105]. Figure 4.2 shows the tree building phase of decision tree algorithm [105].

4.2 During the tree growing phase, a splitting attribute that has the best ability to group the nodes into separate classes is chosen; the splitting of the nodes into separate classes is achieved by reducing the entropies of the new nodes created [115], [125]. Many splitting indexes have been proposed in literature to determine the goodness of best the split of a chosen attribute. The splitting indexes are used in reducing the classification errors during the tree growing phase; the reduction in impurity is known as the information gain (the difference in the change of average entropy or the reduction in entropy). Information gain is a measure that compares the difference of impurity degree when a data is split based on a given attribute [61]. The information gain on a data set (S) that is being split based on an attribute A is given by:

$$\text{Information Gain(S, A)} = \text{Entropy(s)-} \sum_{v \epsilon A} \frac{|S_v|}{|S|} Entropy(S_v)$$

$|S|$ is the number of records

While $|S_v|$ is the subset of $|S|$ for a given value of the attribute v.

Information gain favor attributes with large values which may lead to overfiting.

```
procedure BUILDTREE( Data B)
  If all the data items in B are of the same class or size of class less than threshold
    return
  foreach attribute Ai
    evaluate splits on attribute Ai
  use best split found to partition B into B1 and B2
  GROWTREE(B1)
  GROWTREE(B2)
```

Figure 4.2: Decision Tree Building Phase

The following are some of the other splitting indexes or impurity measures as proposed by [101, 78, 116]. The best value for the impurity measure is obtained when all the data items are grouped into classes or leaf nodes. When data items are skewed, or have large number of unique items, gain ratio splitting measure is preferred to other splitting measures [130].

$$\text{Entropy} = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

The maximum value for the entropy is obtained when all the classes are equally distributed at the given node of the tree and a minimum value is obtained when all the instances of a data set belong to a single class [144, 130]. The disadvantage or bias of this method is that is does not produce an optimal tree when the data set is skewed, and it favors trees with many branching or levels of attributes; it is mainly used with nominal attributes.

$$\text{Gini Index} = 1-\sum_{i=1}^{k} p_i^2$$

p_i is the frequency/probability of the class

k is the number of classes or splits

This measure is more like the Entropy measure because the maximum value for the entropy is obtained when all the classes are equally distributed at the given node of the tree and a

minimum value is obtained when all the instances of a data set belongs to a single class [144, 130]. It has advantage over Entropy since the bias found in Entropy measure is eliminated with Gini index measure, but it is mostly used in binary splitting at the nodes. It does not favor instances with skewed data set or high branching attributes. It takes into account the number and size of branches before selecting an attribute.

$$\text{GainRatio} = \frac{Difference\ in\ Entropy(InformationGain)}{Intrinsic\ Information}$$

$$\text{Intrinsic Information} = -\sum_{i=1}^{k} \frac{N_k}{N} \log[\frac{N_k}{N}]$$

N_k is the number of instances of the child node [144].

Intrinsic Information is the entropy of the record distribution into the nodes of the tree (it gives the information about which branch that belong to a particular record). Unlike the Entropy and Gini index impurity measures favors instances when the data set is skewed. It considers the size and number of the generated child nodes by using the intrinsic information. The best split is one that has a small value of the intrinsic information. The Gain ratio method is used to over the bias identified in the information gain, Entropy and Gini index measures.

4.2.2 Tree Pruning

The Tree pruning phase of decision tree algorithm is used to improve the accuracy of classification and prediction of the algorithm. The sub trees with the smallest error rate are chosen from the decision tree in order to improve accuracy and prevent overfitting (which may be due to noise or too much details in the training data and may result in poor generalization or misclassification error) [78]. There are many pruning algorithms available in literature, but the most popular one is the minimum description length (MDL) proposed by [78, 77]. Other algorithms for tree pruning include cross-validation and separate data set. MDL is used in producing trees that are of small size, with the least amount of coding, with respect to data set and model; thus, the main idea behind MDL algorithm is to encode a tree with the least number of bits [105].

MDL scans a full fledged grown tree in a bottom-up technique unlike tree building which is a top-down strategy. In scanning the tree, the children at a tree node N is pruned, if the minimum cost of encoding it is greater than or equal to the minimum cost of pruning the children of a node directly above it [115]. But if the performance of the pruned tree is the same as the performance of the current tree, the pruned tree is chosen as a result of the Occam's razor (since the pruned tree is less complex than the current tree). Rastogi and Shim, 1998 [105], stated that the total cost of encoding a tree is as stated below:

- The cost of encoding the tree model.
- The cost of encoding the split with respect to the attribute and its value.

- The cost of encoding data items which belong to the same class in the leaf node.

MDL algorithm minimizes the above costs and prunes the tree, then produces a tree with least misclassification error. The Pruning Tree algorithm[105] is shown below;

Pruning Tree Algorithm

procedure PRUNETREE

if A is a leaf Node {

return (Cost(B)) +1

else{

$minCost_1$ = $prunTree(A_1)$

$minCost_2$ = $prunTree(A_2)$

$minCost_A$ = minCost{ Cost(B) +1, $Cost_{Split}(A)$+$minCost_1$+ $minCost_2$}

return $minCost_A$

}

}

$minCost_1$ = minimum cost of pruning subtree (A_1)

$minCost_2$ = minimum cost of pruning subtree (A_2)

Cost(B) +1 = Cost of encoding all data records

$Cost_{Split}(A)$ = Cost of splitting the attribute and cost of encoding the attribute values

4.3 Serial Implementation of Decision Tree Algorithm

Decision tree algorithm can be implemented in a parallel form based on its scalability with respect to the input data. Parallel implementation tends to be scalable, fast and disk resident and can be implemented in computer architecture with many processors [116]. Serial implementation on the other hand is fast, memory resident and easy to understand. In this study, we will focus on serial implementation of decision tree algorithm by Hunt's algorithms and other serial decision tree algorithms that does not obey Hunt's Algorithm (SLIQ and SPRINT). Hunt's method of decision tree construction [51], [103] is as stated below:
Given a training set T of data records denoted by the classes C= $C_1, C_2, \cdots, C_k$

The decision tree is constructed recursively using depth-first divide-and-conquer greedy strategy by the following cases:

- Case1: T contains all the cases that belong to the same class C_j. The leaf node for T is created and it is known by the class C_j

- Case2: T contains cases that belong to one class or more. The best splitting single attribute is chosen, which will test and split T in to a single-class that contains many cases. The split of T gives the subsets of T which are: $T_1, T_2, \cdots, T_n$. The split on T is chosen in order to obtain mutually exclusive results: $O_1, O_2, \cdots, O_n$

  $\forall\ T_i \subset$ T having the result O_i

- Case3: T contains no cases. The leaf created for the decision T has a class from another source which is not T. In C4.5 this is regarded as the most frequent class and is chosen as the parent not of the constructed tree.

With Hunt's method, decision tree is constructed in two phases: tree growth and pruning phases which have been explained in Section II. Most serial decision tree algorithms (IDE3,CART and C4.5) are based Hunt's method for tree construction [126]. In Hunt's algorithm for decision tree construction, training data set is recursively partitioned using depth-first greedy technique, until all the record data sets belong to the class label [51]. The data sets are memory resident and the data sets are sorted at every node in-order to determine the best splitting attribute [116].

One of the disadvantages of serial decision tree implementation is low classification accuracy when the training data is large. In order to reduce the high computational complexity associated with large training data set, the whole training data set is loaded into the memory at the same time which leads to low classification accuracy [126]. This short coming of serial decision tree implementation is addressed by SLIQ and SPRINT algorithm. In serial implementation of SPRINT and SLIQ, the training data set is recursively partitioned using breadth-first technique until all the data sets belong to the same class label and there is one time sort of the data set using list data structure. Also, the training data set is not memory resident but disk resident, which makes data scalability possible. This approach improves the classification accuracy and reduces misclassification errors. The following sections give a review of the commonly used decision tree algorithms based on serial implementation.

Decision trees based on Hunts's algorithm can be classified as classical decision trees which can only be implemented serially. But there have been on-going researches to implement them in a parallel pattern. Peng implemented the parallel version of IDE3 [94]. The disadvantages associated with classical decision tree algorithms are as enumerated below by Podgorelec et al. [97]:

- Handling Noise Data: Classical decision tree algorithms does not always produce decision models with accurate classifications when the training data contain noise or too many details. But C4.5 and enhanced processing technique handles this deficiency.

- Production of Same Type of Decision tree: Given the same training data set and the same condition, the classical algorithm always produces the same tree, instead of producing multiple trees with a flexibility to choose the one that is less prone to error.

- Importance of Error: Different errors arise during application of classical decision tree algorithms, but some errors have higher priority than others and need to be minimized to achieve accurate classification. The errors occur as a result of the decisions made in building the tree which reduce classification accuracy.

4.3.1 IDE3

IDE3 (Iterative Dichotomiser 3) decision tree algorithm was introduced in 1986 by Quinlan Ross [101, 102]. It is based on Hunt's algorithm and it is serially implemented. Like other decision tree algorithms, the tree is constructed in two phases: tree growth and tree pruning.

Data are sorted at every node during the tree building phase in order to select the best splitting single attribute [116]. Algorithm IDE3 uses information gain measure in choosing the splitting attribute. It only accepts categorical attributes in building a tree model [101], [102]. Algorithm IDE3 does not give accurate results when there is too much noise or details in the training data set; thus, an intensive pre-processing of data is carried out before building a decision tree model with IDE3.

4.3.2 C4.5

Algorithm C4.5 is an improvement of IDE3 algorithm, developed by Quinlan Ross (1993) [103]. It is based on Hunt's algorithm and Also, like IDE3 it is serially implemented. Pruning takes place in C4.5 by replacing the internal node with a leaf node; thereby, reducing the error rate [97]. Unlike IDE3, C4.5 accepts both continuous and categorical attributes in building the decision tree. It has an enhanced method of tree pruning that reduces misclassification errors due to noise or too many details in the training data set. Like IDE3 the data are sorted at every node of the tree in order to determine the best splitting attribute. It uses gain ratio impurity method to evaluate the splitting attribute [103].

4.3.3 CART

CART (Classification and regression trees) was introduced by Breiman(1984) [17]. It builds both classification and regression trees. The classification tree construction by CART is based on binary splitting of the attributes. It is also based on Hunt's model of decision tree

construction and can be implemented serially [17]. It uses gini index splitting measure in selecting the splitting attribute. Pruning is done in CART by using a portion of the training data set [97]. CART uses both numeric and categorical attributes for building the decision tree and has in built features that deals with missing attributes [70]. CART is unique from other Hunt's based algorithms, as it is also used for regression analysis with the help of the regression trees. The regression analysis feature is used in forecasting a dependent variable (result) given a set of predictor variables over a given period of time [17] . It uses many single-variable splitting criteria like gini index, symgini and so forth and one multi-variable (linear combinations) in determining the best split point, and data are sorted at every node to determine the best splitting point. The linear combination splitting criteria is used during regression analysis.

SALFORD SYSTEMS implemented a version of CART called CART® using the original code of Breiman(1984) [17]. CART® has enhanced features and capabilities that address the short comings of CART giving rise to a modern decision tree classifier with high classification and prediction accuracy.

4.3.4 SLIQ

SLIQ (Supervised Learning In Ques) was introduced by Mehta et al. (1996) [78]. It is a fast, scalable decision tree algorithm that can be implemented in serial and parallel pattern. It is not based on Hunt's algorithm for decision tree classification. It partitions a training data set recursively using breadth-first greedy strategy that is integrated with a pre-sorting technique

during the tree building phase [78]. With the pre-sorting technique, sorting at decision tree nodes is eliminated and replaced with one-time sort, with the use of list data structure for each attribute to determine the best split point [78, 116]. In building a decision tree model, SLIQ handles both numeric and categorical attributes. One of the disadvantages of SLIQ is that it uses a class list data structure that is memory resident; thereby, imposing memory restrictions on the data [116]. It uses Minimum Description length Principle (MDL) in pruning the tree after constructing it. MDL is an inexpensive technique in tree pruning. It uses the least amount of coding in producing trees that are small in size. It also uses bottom-up technique in pruning a full grown tree [3, 78].

4.3.5 SPRINT

Scalable Parallelizable Induction of decision Tree (SPRINT) algorithm was proposed by Shafer et al.[116]. SPRINT is a fast, scalable algorithm that is designed for large data sets and databases. It generates its tree in a breadth-first approach while other decision tree methods that are based on Hunt's algorithm generate a tree in a depth-first approach. It is not memory-resident, unlike other decision tree algorithms that are based on Hunt's algorithm. It can be implemented in serial or parallel pattern unlike other classification algorithms that are based on Hunt's algorithm which can only be implemented serially. Each class label is linked to the record identification for each value in the attribute list. SPRINT algorithm has two data structures: attribute list and histogram. The attribute list contains the attribute and associated value at each node; also, there is the histogram data structure associated with the attribute list

[125]. The histogram data structure is used to show how the class is distributed at a particular node. Two types of histograms are used for continuous attributes called c-below and c-above [116].

The serial implementation of SPRINT compared to Hunt's based algorithm is similar; however, the difference is that while Hunt's based algorithm builds the tree recursively and determines the splitting point by sorting the data at the nodes with restrictions on data size, since the data are memory resident, SPRINT has no restriction on the input size and uses one-time sort. Thus SPRINT algorithm is disk resident with the training data set [105, 116]. Parallel implementation of SPRINT ensures loading balancing and good data placement at the nodes/classes, by distributing attribute list from the training data set evenly among all N (total number of processors), the fraction of data to be processed by each processor is $1/N$ [116] and each processor determines its best splitting point. This ensures that continuous attributes maintain their sorted order of data at each node. One of the major differences between serial and parallel implementation of SPRINT is the initialization of the histogram data structure. With continuous data, c-below histogram is initialized to zeros, while c-above histogram is initialized to the class distribution using all nodes present. The class histogram is updated for each process of the scan before using the gini-index measure to compute the split point. But with categorical attributes, a single scan process is performed through the attribute list, and the histogram is used after completing the scan; then the gini-index is computed for all the attribute values. The lowest gini-index is chosen [116]. In this study, we will focus on serial implementation of SPRINT decision tree algorithm.

4.4 Serial Decision Tree Algorithm Implementation Statistics

We reviewed about thirty-two articles in order to determine which of the serial decision tree methods is commonly used in practical applications. The outcome of our literature survey is as stated in the following tables below: the articles are represented by the last name of the first author and the year of publication, and the decision tree algorithm used. Table 4.2 show a literature of decision tree algorithms that is implemented serially. Table 4.3 shows the frequency usage of the serial implementation of decision tree algorithms. Table 4.3 shows that IDE3 is the most frequently used classifier, followed by C4.5 and then SPRINT. Algorithm ID3 was one the earliest classifiers but as researchers and scientists discovered its flaws they switched to CART, C4.5 and SPRINT.

4.5 Experimental Analysis

We carried out some experiments using Statlog data sets [79] as shown in Table 4.4. The Stalog data set includes large scale data sets of various disciplines like finance (Australian and German data sets), transportation (Vehicle data sets), science (Shuttle data set) and health (Heart data set). We did a performance evaluation of the decision tree classifiers. Number of records, number of attributes, and class size of the different data sets are varied in-order to

Table 4.2: Literature review of Decision Tree Algorithms

Paper	**Decision Tree Algorithm**
Quinlan, 1983,1993 [100], [103]	IDE3 and C4.5
Shafer et al. 1996 [116]	SPRINT, SLIQ, IDE3 and CART
Hunts et al. 1966 [51]	CLS, C4.5, CART and IDE3
Breiman, 1984 [17]	CART
Fan et al. 2003	Random Tree
Mehta, et al. 1996 [78]	SLIQ
Gehrke et al. 1998 [37]	RainForest
Peng et al. [94]	IDE3
Srivastava et al. 1997 [125]	SPRINT, IDE3 and C4.5
Srivastava et al. 1998 [126]	SPRINT, IDE3, C4.5 and SLIQ
Rastog et al. 1998	PUBLIC, CLS, IDE3, C4.5 and CART
Sattler and Dunemann, 2001 [115]	ID3, C4.5, SPRINT, SLIQ and PUBLIC
Kufrin, 1997 [66]	ID3 and CART
B$\breve{A}$DULESCU (n.d) [6]	Rainforest, IDE3, C4.5 and SLIQ CART and SPRINT
Srivastava and Singh (n.d) [124]	ID3, C4.5, CLOUDS and SPRINT
Sattler and Dunemann, 2001 [115]	ID3, C4.5, SPRINT, SLIQ and PUBLIC
Podgorelec et al. 2002 [97]	ID3, C4.5, CART and OCI
Ling et al. 2004 [71]	C4.5
Du and Zhan, 2002 [28]	ID3 and CART
Pješivac-Grbović et al. 2006 [96]	C4.5
Wen et al. 2008 [143]	CART and C5.0
Xu et al. 2006 [146]	IDE3 and $IDE3^{+}$

determine their effect on the performance of each classifier. Tables 4.5, 4.6, and Figures 4.3, 4.4 show the result of the analysis.

4.5.1 Experimental Results

Figure 4.4 shows that for all the classifiers, the execution time increases as the number of records increases. The steady portion of the graph is the effect of varying the number of attributes of the classifiers. Also Figure 4.4 shows that the execution time (time to build the model) of the classifiers decreases as the attributes of the classifiers increase and become

Table 4.3: Frequency use of Decision Tree Algorithm

Decision Tree Algorithm	Frequency Usage
CLS	9 %
IDE	68 %
$IDE3^{+}$	4.5 %
C4.5	54.55 %
C5.0	9 %
CART	40.9 %
Random Tree	4.5 %
Random Forest	9 %
SLIQ	27.27 %
PUBLIC	13.6 %
OCI	4.5 %
CLOUDS	4.5 %
SPRINT	31.84 %

steady at some point due to the change in the number of records of the data sets and class size change. Table 4.5 shows that SPRINT classifiers have the fastest execution time among all the classifiers, irrespective of the class size, number of attributes, and records of the data sets' volume. Algorithm C4.5 closely follows SPRINT algorithm in performance. The table also showed that the execution time for IDE3 is faster than CART, but CART is preferred by researches and scientists as it handles both categorical and continuous attributes, while IDE3 does not handle continuous attributes. Table 4.6 shows that SPRINT classifier has the highest classification accuracy among all the classifiers; the classification performance is followed by C4.5. Compared to other classifiers, the class size, attribute number, and record number do not affect the classification accuracy of SPRINT and C4.5. The classification accuracy of the IDE3 and CART classifiers depends to a large extent upon the class size, attribute number, and record number of the data sets. As shown in Table 4.6, for a large data set (shuttle data set), the classification accuracy of IDE3 is better than that of CART as ID3 has a high

accuracy for large data that have been pre-processed (noise and outliers removed) and loaded into the memory at the same time. But for other data sets (Vehicle, Australian, German and Heart) that are not too large (small-medium data sets), the classification accuracy of CART is more than that of IDE3.

Table 4.4: Statlog Datasets

Dataset	Category	No. of Attributes	No. of Classes	No. of Records
Australian	Credit Analysis	14	2	690
Shuttle	Space Shuttle Radiation	9	7	43499
German	Credit Analysis	24	2	1000
Heart	Heart Disease Screening	13	2	270
Vehicle	Vehicle Identification	18	4	753

Table 4.5: Execution Time to build Model

Dataset	IDE3	CART	C4.5	SPRINT
Australian	0.08secs	11.41secs	0.02secs	0.02secs
Shuttle	1.48secs	38.31secs	0.17secs	0.15secs
German	0.03secs	2.17secs	0.06secs	0.04secs
Heart	0.03secs	0.61secs	0.1secs	0.03secs
Vehicle	0.03secs	1.64secs	0.1secs	0.02secs

Table 4.6: Classification Accuracy

Dataset	IDE3	CART	C4.5	SPRINT
Australian	71.5 %	85.4 %	84.2 %	85.8 %
Shuttle	99.2 %	94 %	98.00 %	99.63 %
German	32.1 %	70 %	69.2 %	70 %
Heart	35.2 %	56.67 %	76 .7 %	80 %
Vehicle	54.3 %	65 %	66.5 %	67 %

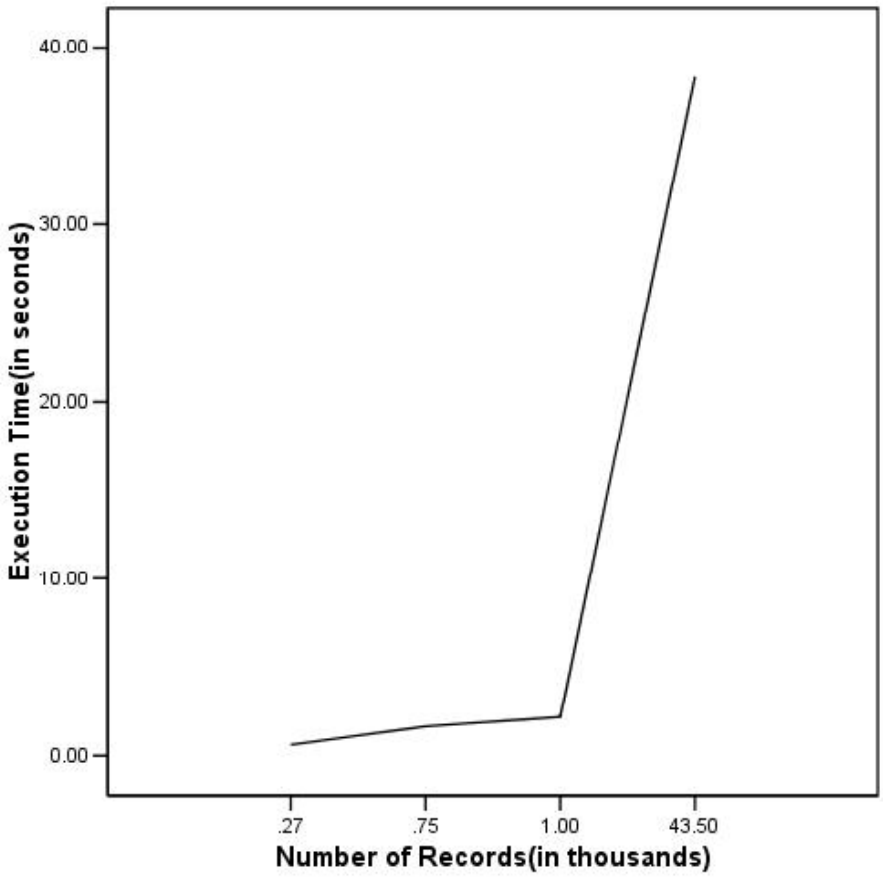

Figure 4.3: Execution time and Number of Records

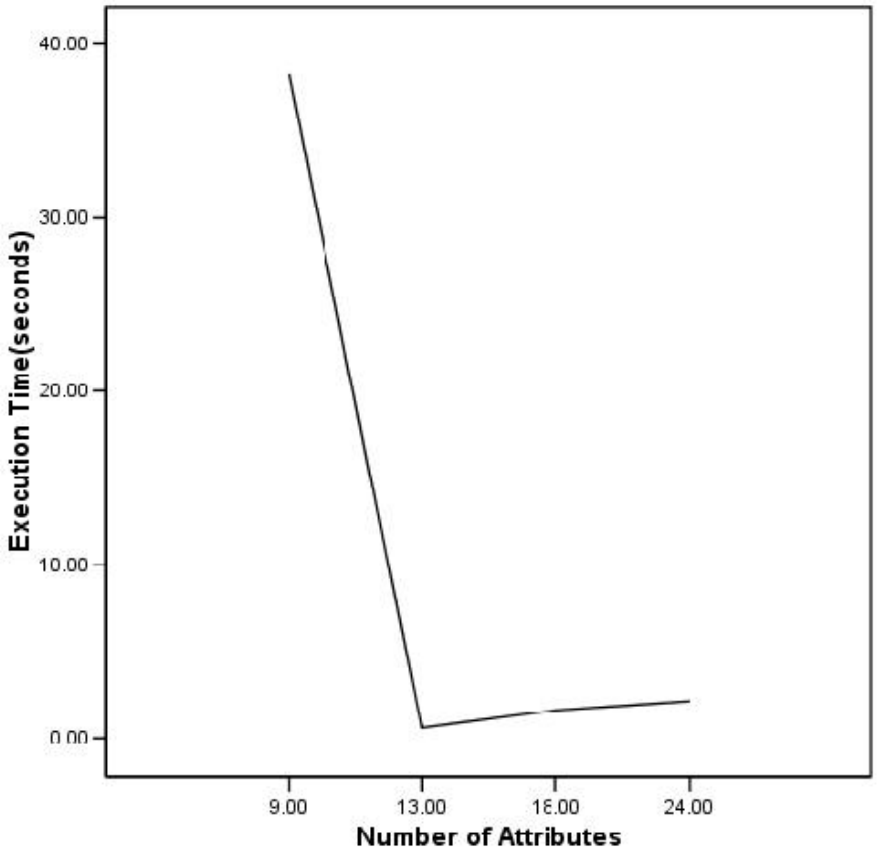

Figure 4.4: Execution time and Number of Attributes

4.6 Predicting Model

The Oxford English Dictionary [121] defines *model* as "A simplified or idealized description or conception of a particular system, situation, or process, often in mathematical terms, that is put forward as a basis for theoretical or empirical understanding, or for calculations, predictions, and so on.; a conceptual or mental representation of something." Modeling can be described as an inseparable part of all intellectual activities, especially scientific activities which involve generation of abstracts, concepts, models, and so on [120]. Model in science can also be said to be an abstraction of the real problem, process, object, method, phenomena, and so forth. Modeling also, involves the application of special tools, techniques to generate outputs that are reliable, and can be verified and validated, consistent and correct [120]. Models are generally of two types: predictive and explanatory [47]. While explanatory models give a summary of the data description in terms of a particular domain, predictive models give an insight into the unseen or future case/data, given the past and present case/data. In this study, we will focus on predictive models for decision tree classification algorithms.

4.6.1 Predicting Model Complexity:

The complexity of a prediction model like classification strategy is determined using Vapnik-Chervonenkis dimension (VC dimension) by Vapnik and Chervonenkis [137]. The VC dimensions is used to know the complexity/capacity of a classification algorithm, and determine over-fitting in the algorithm. VC dimension of a given class gives the maximal number

h (set of data points) which can be shattered [16, 137]; thus, h is the maximum number of points that can arranged in such a way that the classification algorithm can shatter them (Moore, 2001). A classification algorithm D is said to shatter a set of data points X_0, X_1, X_2, ... X_{n-1}, if for every training data of the form (X_0, Y_0), (X_1, Y_1), (X_2, Y_2), ... (X_{n-1}, Y_{n-1}) where Y_0, Y_1, Y_2, ... Y_{n-1} are the class labels, there exists a constant with some value β that has a zero training error [137]. It is assumed that all the points in the training data set are independent and identically distributed (i.i.d). A classification model with zero training error has generation problems and it is more complex than one without training error.

The VC dimension can also, be used to calculate the upper bound for test error in classification algorithm. Vapnik and Chervonenkis (1971) [137] gives the upper-bound of the test error with a probability of 1-η as

$$T1 = T2 + \sqrt{\frac{h(\log(2N/h) + 1) - \log(\eta/4)}{N}} \tag{4.3}$$

h = VC dimension, η = confidence level of value $0 \leq \eta \leq 1$, N = size of the training data set.

T1 = Test error and T2 = Training error.

4.6.2 Model Evaluation:

After building and constructing a model, the model is evaluated based on the objectives used in building it. In addition to fulfilling its objectives, the following criteria are used in evaluating a given model:

- Consistency: A model has to be consistent (fit) with the empirical data (test data) used in evaluating and testing [120]. A model that does not fit data will be rejected or subjected to review/modification.

- Prediction of future observation: A good model should predict class labels of data sets that are not used in the training or test data set.

- Complexity: The VC dimension of a good model should not be high in order not to shatter all the data points in the training data set. If the complexity of a model is low, then a good generation is possible for the model. Thus a model should be easy to use, especially with other models.

- Accuracy: A good model should have a reasonable degree of confidence level in predicting observations.

4.7 Summary

Decision tree induction is one of the classification techniques used in decision support systems and machine learning processes. With the decision tree technique, the training data set

is recursively partitioned using depth-first (Hunt's method) or breadth-first greedy technique [116] until each partition is pure or belongs to the same class/leaf node [51, 116]. Decision tree model is preferred among other classification algorithms because it is an eager learning algorithm and easy to implement. Decision tree algorithms can be implemented serially or in parallel. Despite the implementation method adopted, most decision tree algorithms in literature are constructed in two phases: tree growth and tree pruning phase. Tree pruning is an important part of decision tree construction as it is used in improving the classification/prediction accuracy by ensuring that the constructed tree model does not overfit the data set, [78]. In this study, we focussed on serial implementation of decision tree algorithms which are memory resident, fast, and easy to implement compared to parallel implementation of decision tree that is complex to implement. The disadvantages of serial decision tree implementation is that it is not scalable (disk resident) and its inability to exploit the underlying parallel architecture of computer system processors.

Our experimental analysis of performance evaluation of the commonly used decision tree algorithms using Statlog data sets [79] shows that there is a direct relationship between execution time in building the tree model and the volume of data records. Also, there is an indirect relationship between execution time in building the model and attribute size of the data sets. The experimental analysis also shows that SPRINT and C4.5 algorithms have good classification accuracy compared to other algorithms used in the study. The variation of data sets' class size, number of attributes, and volume of data records is used to determine which algorithm has a better classification accuracy between IDE3 and CART algorithms. In the fu-

ture, we will perform experimental analyses of commonly used parallel implementation tree algorithms and then compare it to that of serial implementation of decision tree algorithms and determine which one is better based on practical implementation.

Chapter 5

Churn Analysis and Management

Churn phenomenon cuts across all sectors of business. This phenomenon includes but is not limited to, financial, telecommunication, retail, education, and so on. Market liberalization of once closed markets, government regulations, privatization of government held monopolies in some industries, improvement in technology, provision of goods/services at discount rates, provision of effective and efficient services, and branding of goods/products etc, has provided competitive environments for organizations. The competitive market environments makes it possible for customers to become churners thus, switching loyalty among different organization/firms in the same line of business [72, 92]. In this study, we apply an enhanced decision tree algorithm to predict and detect the possibility that a customer will churn and present some techniques that will prevent churning.

5.1 Churn Analysis

Customer churn in a business environment can be described as the process by which a customer discontinues the use of a product or service and continues the use of a similar product or service with another company. It can be described as the process of customer turnover [106] or losing a customer or business. Churn rate as shown in (5.1) for a given organization can be viewed as the percentage or number of customers lost compared with the total number of active customers [133, 52]:

$$Churn\ Rate = \frac{customers\ \ lost}{Total\ number\ of\ active\ customers} \tag{5.1}$$

Customer churn can be unavoidable, voluntary or involuntary [85]. A churn is unavoidable if the customer dies or is in a mentally unstable state. It is voluntary if a customer willingly terminates the services or discontinues the use of a particular product. The discontinuation of use of a product or service may be due to loss of jobs, missed payments, mortgage prime loans, and so on. In this chapter, we will focus on voluntary customer churn in which customers willingly switch between competitors in the same line of business environment in order to gain a monetary advantage, improved service, or received better quality products. Customer churn is highly related to customer retention and loyalty; a good customer rela-

tionship strategy is needed for customer retention and loyalty [89]. Customer loyalty can be calculated using the equation as shown in (5.2) [89].

$$Customer\ Loyalty = 1 - Churn\ rate \tag{5.2}$$

Customer churn analysis is used in identifying customer groups that are at risk of defecting and the customer lifetime value (CLV) of a business [85]. The life time value of a customer is the total net income/benefit of the customer to the business over the customer's lifetime with the business [89]. The customer churn segments are detected using the following techniques [122]:

- Churn Trend Analysis: This trend uses the sales trend of the amount spent by customers on goods and services. It identifies customers in terms of sales amount, customer numbers, values of goods, and services consumed, over a period of time. The sales trend can be discovered using time-series analytical tools.

- Churn Profiling: This techniques identifies risky segments based on geographic, demographic and psychographic groups. Customer profiling is used in identifying the segments that are most likely to churn and the reasons for defecting.

- Churn Prediction: This technique builds a model based on churn analysis (churn trend and profiling). The model identifies the possibility of a customer defecting and the

segment of the customer base at risk of churning based on scoring indices as discussed in the sections following.

5.1.1 Churn Indices

The main aim of churn analysis is to understand and predict customers who are in the risky group of defecting to a competitor's business and then to advise management on ways to retain those customers that are profitable [89]. Customer churn prediction varies with the type of organization and customer's ability to churn can be determined by scoring churn indices [122]. In the retail supermarket chain industry, segmentation of customer records is used to determine the buying pattern or behavior of the customer over a given period of time which is a churn index; then any deviation from the discovered pattern can likely be described as a churn behavior on the part of the customer. Switching (churn index) of subscribers from one provider to another can be taken to be churn in the telecommunication or internet service industry [89, 129].

A record showing that a student did not graduate(churn index) after a specified number of years or that the student "dropped-out" could likely indicate that the student transferred to another school, the action implying that the student has churned in the education sector. In the finance industry for a company issuing credit cards to its customers, the indication that the customer has churned (start using another credit card from another company) is the decline in transactions (churn index) or closure of the credit card account by the customer. The customer churn prediction on a target industry will be based on the data that are available.

An analysis by [52, 122, 133] of customers in the telecommunication industry shows the following as the possible churn factors:

- Customer Location: A change in customer residence or work area could indicate possible churning to a business competitor. Also, change in customer status, like increase in income or lifestyle, can be a reason for churning as the customer may want to belong to the elitist segment of society.

- Low Quality Service or Goods: If customers are not satisfied with the quality of goods or services, they will likely defect to a business competitor who will meet their demand.

- High Cost of Goods and Services: If customers discover that goods and services are highly priced compared to those of business competitors, they may likely defect to take advantage of the low prices depending on the elasticity of the goods.

5.1.2 Churn Management

Churn management is part of the customer relationship management (CRM) a business entity adopts in order to retain its customers [141]. It is far more expensive to attract new customers than to retain existing ones [122], thus the need for effective churn management. The churn management technique is used to determine the probability that a customer will churn and the possibility that the customer will remain loyal. It includes a predictive and a recommender system that will build customer loyalty and retentive strategy and at the same time determine customer profitability. Churn prediction is used to determine effective churn

management/strategy that will optimize profitability and enhance customer relationship management [147]. Figure 5.1 shows the process of churn prediction and management. In this research we use an enhanced decision tree algorithm as described in the sections below to predict and determine the possibility that a customer will churn and suggest appropriate churn management/customer relationship management techniques to address it.

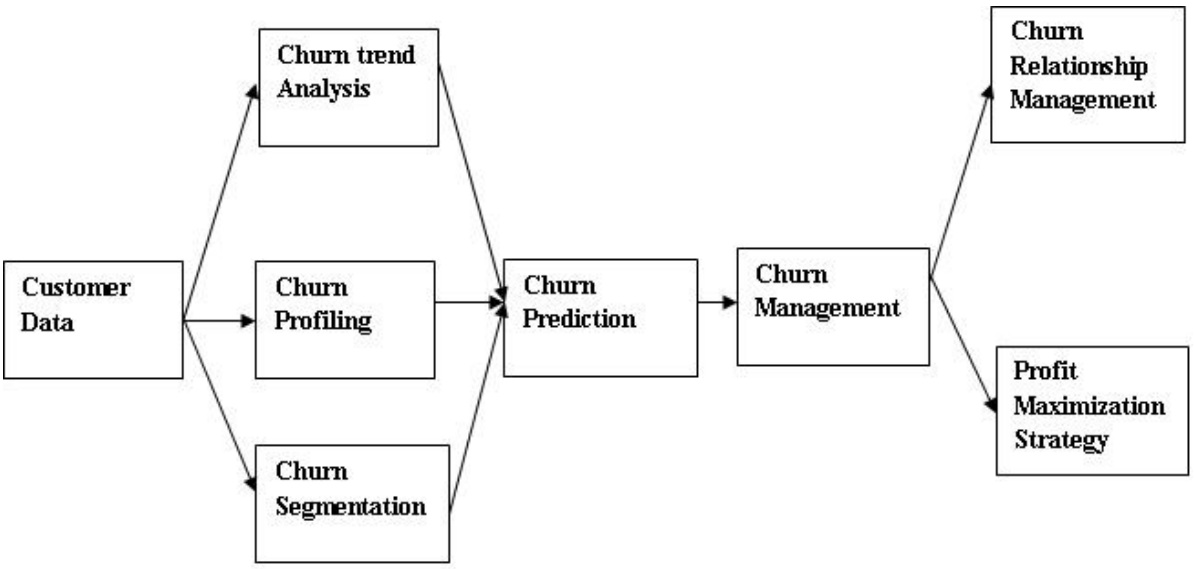

Figure 5.1: Framework of churn prediction and management

5.1.3 Churn Model

The Churn model is generally of two types: one that considers the customer's leaving the firm permanently and one in which the customer defects to a business competitor in the same line

of business et al. [40]. The churn models are Hazard and Markov models. The Hazard model looks at customers' defection as permanent, such as the death, loss of income, relocation of residence, and change of job. It uses probability measure based on the failure time of customer (period of time customer fails to patronize the firm) to determine the possibility that a customer will churn [40].

The Hard Model known as accelerated failure time (AFT) [60] is as given below:

$$\ln(t_j) = \beta_j X_j + \sigma\mu_j \tag{5.3}$$

t= purchase time of customer j.

X= covariants, β and σ are constants which can have different values.

The Markov model is used to describe the probability that a customer defects to competitors's business; thus, the customer transits from one state to another [60]. The model is used in predicting the transition probabilities of a customer with a particular business at any particular time (being in a particular state). Pfeifer and Carraway [95] used the transition probability to calculate CLV as given below:

$$V^{'} = \sum_{t=0}^{T} [(1+i)^{-1} P]^t R \tag{5.4}$$

V' = CLV of the transition states

P = Probability matrix of the transition which can be constant over a given period of time. It could also be said to be the switching probabilities [60], [110], [68]. The R is margin (reward vector),which can be said to be constant over a given period of time; it is the total volume of items purchased and the real possibility of buying a particular item at any given time. Pfeifer and Carraway [95] defined transition states of the Markov model based on the recent purchase or activities of the customers. It could be defined based on the additional states for both the new and existing customers [60].

Chapter 6

Implementation and Results

6.1 Proposed Algorithm

Decision tree algorithm first builds a decision tree and then prunes subtrees from the decision tree in order to improve accuracy and prevent overfitting (which may be due to noise or too much details in the training data and may result in a poor generalization or a misclassification error) [51]. SPRINT decision tree algorithm uses entropy or gini index impurity measure to calculate the goodness of the best split, but our enhanced decision tree algorithm uses gain ratio to calculate the goodness of the best split [130]. The Entropy/gini index impurity measure tends to favor data items that are large and distinct but churn rate is usually slow in nature. Thus the entropy/gini index measure will not determine the correct goodness of the best split. In order to overcome the short-comings of entropy/gini index, gain ratio impurity measure is used. Decision trees algorithm does not give accurate results when the data are

skewed, especially in churn prediction as the algorithm works by splitting data attributes into smaller groups and then use, the dominant group which represents 50% of the segment population [122]. Also, there is a low rate nature of churning which leads to inaccurate decisions. Thus, accurate churn prediction will not be possible when decision tree algorithm alone is used in the prediction/classification [122].

In order to avoid the error associated with decision tree algorithm, we propose an improved decision tree algorithm that will perform a churn trend analysis by first clustering (K-Means clustering method may be used) the customer data record which will produce a customer churn profile based on the following groups or patterns: geographic, demographic and psychographic attributes. Clustering algorithms will also, explore the data to discover customer behavior or patterns; the exploration gives a good view of the data structure such as the purchasing power of the customer in a retail store business. The result of data exploration will enable the management to adopt good marketing strategies to maintain customer loyalty. The enhanced SPRINT algorithm is then applied to the customer record after the data has been clustered. The possibility of customer defection is detected by churn scoring. The training data will be used to train the classification model (decision tree). The test data will be used in validating the model. The clustered data will be validated using clustering validation algorithms. We compared the result of experimental analysis of cluster validity measures based on proximity measure with that of cluster measures obtained from literature survey of the measures; then the best cluster validity measure are used in the validation.

The clustering of the data will reveal the customer demographics, geographic, psychographies and behavior/patterns towards the target organization, service, or product. The pattern displayed by the segmented (clustered) data will be used to predict customers who are likely to churn. The churn rate can then be determined using the total customer base and the customers who churned. Thus, the data mining technique will be used in customer churn detection. Customer churn detection as explained above involves churn trend analysis (using sales trend analysis to determine churning pattern), customer churn profiling (determine churn risk groups based on demographics, geographic and psychographies), and prediction of future churn rate of the customers. The customers' records, segmented records, and other external information from business/market environment will be used in creating a data warehouse. Figure 6.1 shows our proposed decision tree model. Figure 6.2 shows a frame work of the model.

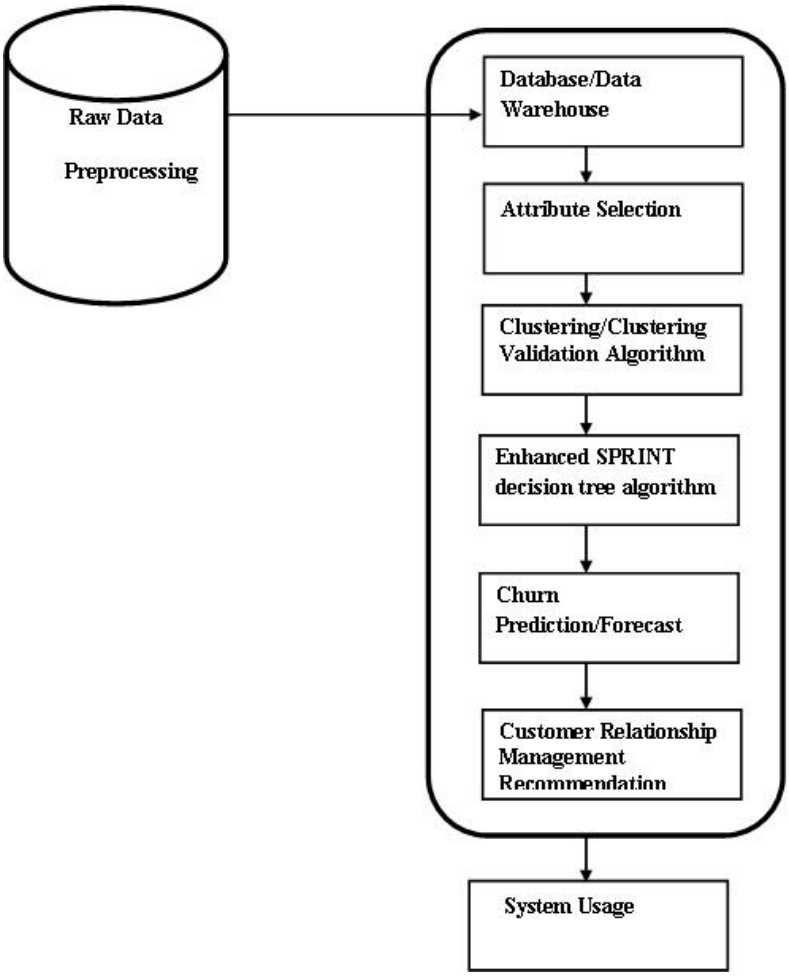

Figure 6.1: The proposed Model

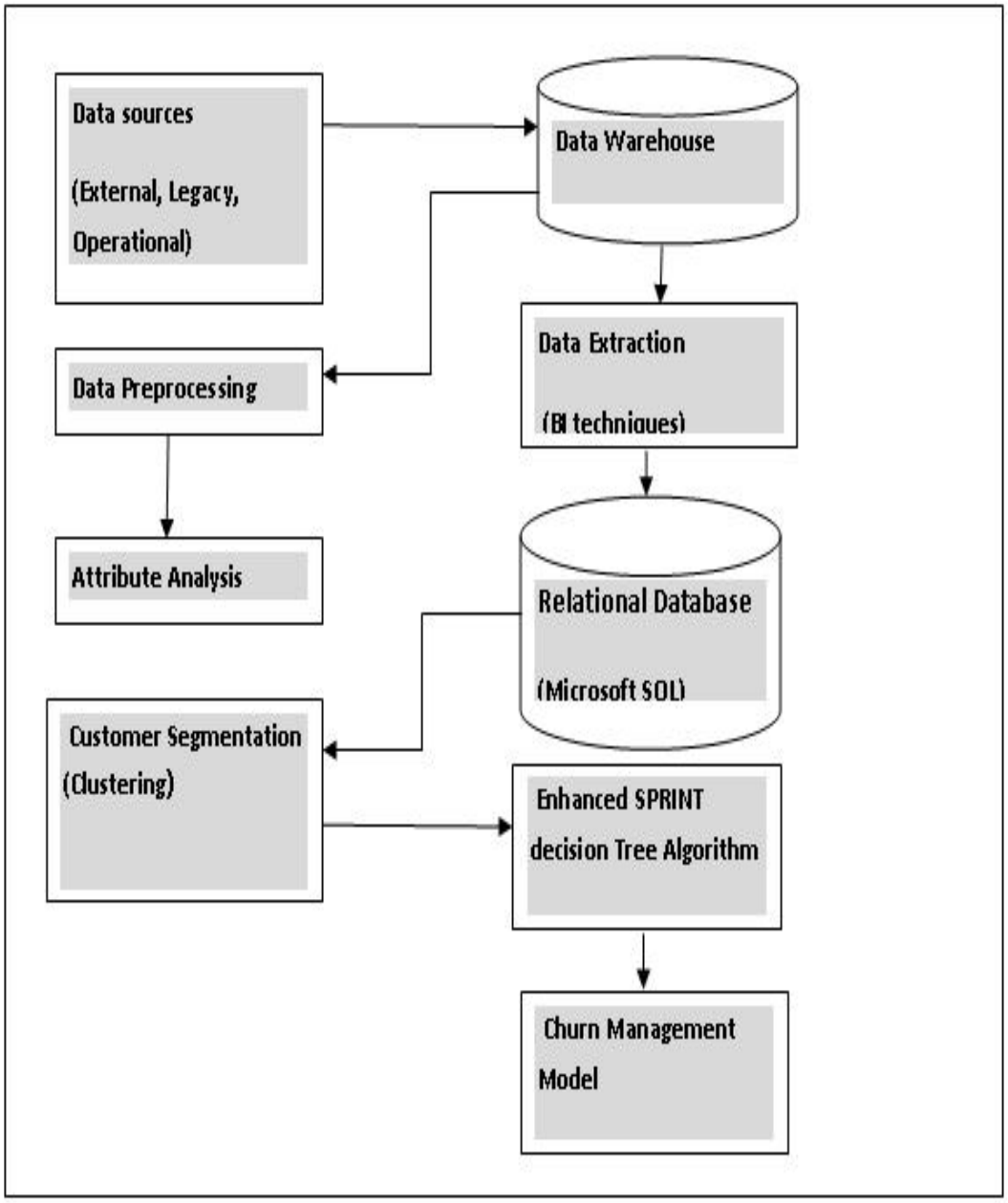

Figure 6.2: The proposed Model Framework [50]

6.2 Algorithm Complexity of Decision Tree

Decision tree construction is a greedy algorithm that recursively partitions a training data until each partition contains record which is of the same class or a leaf node [116] or until there are no more nodes to split [88] in a top-down pattern and then prunes the full grown tree using a bottom-up approach. The tree pruning ensures that overfitting of the training data is avoided and noise in the data is eliminated. The greedy nature of the algorithm is that at each node the best split test decision is made. The best split decision at each node is a local optimal choice [88] that produces suboptimal trees [39]. The optimality of a decision tree is determined by its prediction/classification accuracy and the size/depth of the tree [88]. The smaller or shallower the tree, the more computationally efficient the tree, and the higher the classification/prediction accuracy. Also, an optimal tree ensures that the number of steps required to predict/classify an unknown data is minimized by building a tree that is small in size or shallow in depth [53]. Constructing an optimal binary tree is intractable; it is an NP (nondeterministic polynomial time)-complete problem which requires a heuristic approach. Thus a polynomial time solution for optimal tree construction is very unlikely, that is P (polynomial time)$\neq$N [53, 87]. A search or decision problem is NP-Complete when other search or decision problems are reducible to it [22]. Our proposed model improves the optimality of a SPRINT decision tree algorithm by increasing its prediction/classification accuracy as shown in section 6.3.1. This optimality is achieved by enhancing its splitting attribute

and segmenting the training by using the clustering algorithm before applying decision tree algorithm.

6.3 Implementation

We implemented our enhanced decision model by first clustering the data using K-means algorithm. We modified the SPRINT decision tree algorithm by changing the splitting index from gini index to gain ratio; this addresses the skewness of the data. The SPRINT decision tree algorithm is a recursive partitioning algorithm which is used in variable/attrubite reduction; thus, the issue attribute selection is eliminated [99]. We also gain leverage on the query capability of SQL back-end database. The use of a database engine in implementing our algorithm makes it to be scalable to a large database set which enabled us to obtain enough statistics on the data and on the model. The database engine also ensures that data used in building the model is disk resident, ensuring efficient memory management and selection of the required attribute. We validated the results of the clustering algorithm by developing a system that compares the clustering indices. We equally perform a literature survey of all the clustering indices and compare it with our survey result and the results we obtain from the system we developed. We chose the best clustering index from our comparison and used it to validate the result from the clustering algorithm.

6.3.1 Data Preparation and Algorithm Processing

We collected about $28,867$ retail records from Microsoft retail data mining records (Table 6.1) [80], about $5,000$ of financial bank data records from De Paul University, Chicago USA, data resource (Table 6.2) [84], and about 270 database instances from V.A. Medical Center, Long Beach, and the Cleveland Clinic Foundation (Table 6.3) [26]. In Table 6.1, the records are pre-processed and analyzed; then unwanted attributes like region, ID and commute distance were removed. Since our proposed algorithms can handle numerical attributes, there is no need to discretize other attributes. The attributes used in predicting bike buyer class from the retail data are marital status, gender, yearly income, children, education, occupation, home owner, cars and age. These attributes are used in determining which of the customers will be a bike buyer (positive classification) and those that will not buy bikes (negative classification).

The customers who did not buy bike churns; thus, our algorithm detects customer churn in the organization. In the financial bank record data set as shown in Table 6.2, the ID attribute was removed, and other numerical attributes were not discretize. The data sets are used in predicting the bank customers that will invest in personal equity plan (pep) class. All other attributes used in the prediction are shown in Table 6.2. Customers who did not invest in the investment plan are assumed to have churned; that, is they are likely to close their accounts with the bank or invest in the plan with another financial institution. Table 6.3 contains patients' data records. It is used to predict patients that will likely have heart disease using the patients' medical history records. It also determines the patients that bear

Table 6.1: Retail Data

Attributes/Fields	Description
ID	Record identification
Marital Status	Single or Married
Gender	Male or Female
Yearly Income	Total income for the year(numeric)
Children	Number of children(numeric)
Education	Bachelor,high sch.graduate,partial college
Occupation	Clerical,professional,skilled,management
Home owner	Yes or No
Cars	Total number of cars(numeric)
Commute Distance	Distance Commuted(numeric)
Region	Europe, Pacific,North America

Table 6.2: Financial (bank) data

Attributes/Fields	Description
ID	Record identification
Age	customer's age(numeric)
Sex	Male or Female
Region	inner_city/rural/suburban/town
Income	The income of customer(numeric)
Married	Marital status of customer(YES/NO)
Children	The number of children for a customer(numeric)
car	does customer have cars(YES/NO)
sav_acct	If customer have saving account(YES/NO)
current_acct	If customer have current account(YES/NO)
mortgage	If customer have a mortgage plan(YES/NO)
pep	If customer invested in personal equity plan (YES/NO)

the risk of having heart disease in the future. These risky group of patients are those that churn (patients with no heart disease) that stopped being patients of the hospital since the hospital specializes in treating patient with heart disease. The database instances in Table 6.3 have been pre-processed and cleaned. It contains no missing values of the attributes.

Table 6.3: Medical (Heart Disease) data

Attributes/Fields	Description
Age	Patient's Age
Sex	Patient's sex (1=male,0=female)
Chest pain type	Chest Pain Type (values: 1,2,3 and 4)
Blood Pressure	Resting Blood Pressure (measured in mm Hg)
Cholesterol level	Serum Cholesterol in mg/dl
Blood Sugar	Fasting Blood Sugar $<$ 120 mg/dl (1=true,0=false)
Electrocardiographic Results	Electrocardiographic Results (0=normal) (1=ST-t abnormality,2=left ventricular hypertrophy)
Maximum Heart Rate	Maximum Heart Rate achieved
Chest pain	Exercise Induced angina (1=yes, 0=no)
Oldepeak	ST depression induced by exercise relative to rest
ST_slope	The slope of the peak exercise ST segment (Values: 1=upssloping,2=flat,3=downsloping)
vessels	The number of major vessels (0-3) colored by fluoroscopy
Thalassemia (Thal)	thal: 3=normal, 6=fixed defect, 7=reversible defect
Heart Disease	Predicted attribute (1=absence of heart disease, 2= presence of heart disease)

Table 6.4: Churn prediction of retail data

Name	Marital Status	Children	Income	$\cdots$	Predicted Bike Buyer
Candidate1	Married	0	\$90000	$\cdots$	Yes
Candidate2	Single	3	\$60000	$\cdots$	No
Candidate3	Married	3	\$60000	$\cdots$	Yes
Candidate4	Single	0	\$70000	$\cdots$	No
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Table 6.5: Churn prediction of financial data

Name	Age	Sex	Income	···	Predicted Pen Investment
Candidate1	23	Male	$18766.9	···	Yes
Candidate2	30	Male	$9915.67	···	No
Candidate3	45	Female	$21881.6	···	Yes
Candidate4	50	Male	$46794.4	···	Yes
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮

Table 6.6: Churn prediction for heart disease

Name	Age	Sex	Chest pain	···	Predicted Heart Disease
Patient1	70	1	4	···	1
Patient2	67	0	3	···	1
Patient3	57	1	2	···	1
Patient4	64	1	4	···	2
Patient5	74	0	2	···	1
Patient6	65	1	4	···	2
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮

6.3.2 Implementation The Proposed Model using Weka

Weka Data Mining Software in Java [144] is enhanced in-order to implement the proposed algorithm. Weka in its current form is not suitable to implement the algorithm; thus, Weka's source code is modified to accommodate the implementation of the proposed algorithm. The java code generated by the modification of Weka source code is given in Appendix A. The following steps are taken in-order to modify the Weka software:

- Software Environment: Java JDK 1.5 or later for Weka 3.5.x or later. Also, NetBean IDE 6.5 or latter is used for the Java environment.

- Setting up the database connections: The database use is MS SQL Server 2005. Weka DatabaseUtils.props package for MS SQL Server 2005 is DatabaseUtils.props.mssqlserver2005. DatabaseUtils.props.mssqlserver2005 properties that is amended for the proposed algorithm are:

  jdbcDriver and jdbcURL

  The required jdbcDriver is com.microsoft.sqlserver.jdbc.SQLServerDriver

  while the required jdbcURL is jdbc:sqlserver:server.my.domain:1433

  The amended DatabaseUtils.props file for Microsoft SQL server 2005 is

  as stated below:

jdbcDriver

jdbcDriver= com.microsoft.sqlserver.jdbc.SQLServerDriver

jdbcURL

jdbcURL=jdbc:sqlserver:/localhost:1433

Text

char=0

tinytext=0

text=0

varchar=0

longvarchar=0

binary=0

varbinary=0

longvarbinary=0

blob=0

mediumtext=0

mediumblob=0

longtext=0

longblob=0

```
# Number types
    bit=1
    numeric=2
    decimal=2
    float=2
    double=2
    tinyint=3
    smallint=4
     short=5
    integer=5
     int=5
     mediumint=5
    bigint=6
     long=6


# Data Types
     real=7
     date=8
     time=8
     timestamp=8
```

datetime=9

mappings for table creation

$CREATE\_STRING = TEXT$

$CREATE\_INT = INT$

$CREATE\_DOUBLE = DOUBLE$

database flags

checkUpperCaseNames=false

setAutoCommit=true

createIndex=false

flags for loading and saving instances using DatabaseLoader /Saver

nominalToStringLimit=50

$idColumn = auto\_generated\_id$

- Setting up the TCP/IP Connection: In order to establish a connection between the Weka package and the SQL server, Microsoft SQL Jdbc driver has to be installed and configured the classpath has to be set to reflect the jar file(sqljdbc). Also, the TCP/IP connection has to be enabled in the SQL server 2005.

- Adding the Enhanced Sprint algorithm class:

  A class Sprint.java is created with the Weka directory

  $< path >$/weka/classifiers/trees/Sprint.java The GenericObjectEditor property of the Weka is edited to include the Sprint class in the appropriate class super class interface for usage in the GUI for the Weka system. The Sprint.java is a modified with the following additions:

  The Instances class is modified to enable files to be read in an incremental mode instead of reading the whole file into the memory at the same time. This reading of the file makes the Sprint algorithm to be scalable as all the data are not resident in the memory at the same time. This reading is done by adding the code below:

```
BufferedReader mato = new BufferedReader(new FileReader("/< path >/filename"));
 ArffReader arff = new ArffReader(reader, 1000);
Instances data = arff.getStructure();
data.setClassIndex(data.numAttributes() - 1);
Instance inst;
while ((inst = arff.readInstance(data)) != null) {
data.add(inst);
 }
```

ArffReader is the BufferedReader class.

BufferedReader is a java class used in reading character-input stream from a text/file.

It buffers the characters from the file/text so that the characters are read line by line, instead of reading the whole text/file into the memory at the same time.

6.3.3 Implementation Results

We did a comparative analysis of classification of our proposed model with other decision tree algorithms in terms of percentage of class prediction and other classification accuracy measures. The experimental analysis was done repeatedly and consistent results were obtained. The results of the experimental analysis are as stated in section 4.1. Tables 6.7, 6.8, 6.9, 6.10 and 6.11, show the comparative analysis of our algorithms and other decision tree algorithms.

Table 6.7 and 6.8 show the percentage of class prediction while Tables 6.10,6.11, 6.12 and 6.13 show the classification accuracy measures of precision, recall, F-measure and ROC. Figure 6.4 shows the percentage of class prediction for the bike buyer. Table 6.4 shows the sample of the churn prediction for bike buyers from the retail data (bike buyer is the predicted class).

The customers that do not buy bikes are mostly likely to churn. Exploration and forecasting of the customer records shows that customers that are married with or without children are most likely to buy bikes while those that are single and without children are less likely to buy bikes and thus more likely to churn. Table 6.5 also, shows a sample of churn prediction

for bank customers that will invest in the personal equity plan (pep is the predicted class), the customers that did not invest in the plan are most likely to churn. Also, exploration and forecasting of the customer records show that married customers who have mortgage plans and earn higher income than other customers are likely to invest in the plan. The result of this analysis will assist management to put in place programs that are customers centric in order to retain their customers and achieve customer loyalty. Table 6.6 shows the churn prediction for patients that are suffering from heart disease. Heart disease is the predicted class; the value of 1 signifies the absence of heart disease while the value of 2 signifies the presence of heart disease in the patients.

Analysis of the prediction results shows that following attributes are dominant in predicting a patient with a heart disease: Thal is a blood disorder that is inherited. The blood disorder is responsible for the abnormal shape of the Thal. Thal disorder leads to destruction red blood cells and anemia. Thal with values (6=fixed and 7=reversed), Vessels with values(1,2, and 3) , chest pain type with value 4, chest pain with value (1=yes), oldpeak with values between the range (2.06-6.2), maximum heart rate with values between the range (71.0-136.0) and sex with value(0=female). Thus, patients with this medical record history(stated attributes) without heart disease are likely to churn, but the hospital management will require them to be on a treatment/management plan since they are at risk of having heart disease in the future if their case is not well handled. This result shows the use of churn management model in preventing churning pattern of customers. Figure 6.3 shows the decision tree prediction for patients with heart disease. The numbers in parenthesis specify the total number attributes

Table 6.7: Comparative analysis of classification accuracy using retail data

	Correctly Classified	Incorrectly Classified	Unclassified
Proposed Algorithm	91.02%	8.98%	none
C4.8(J48)	86.30%	11.97%	1.7%
Random Tree	87.79%	12.21%	none
Random Forest	89.67%	10.33%	none
Simplcart	90.5%	9.4%	none

Table 6.8: Comparative analysis of classification accuracy using Medical (Heart disease) data

	Correctly Classified	Incorrectly Classified	Unclassified
Proposed Algorithm	80%	20%	none
C4.8(J48)	79.2%	20.8%	none
Random Tree	71.11%	28.89%	none
Random Forest	78.15%	21.85%	none
Simplcart	77.78%	22.22%	none

that are correctly classified and those that are not correctly classified (correctly classified / incorrectly classified).

Table 6.9: Comparative analysis of classification accuracy using Financial Data

	Correctly Classified	Incorrectly Classified	Unclassified
Proposed Algorithm	68.67%	31.3%	none
C4.8(J48)	67.67%	32.3%	none%
Random Tree	56.7%	43.3%	none
Random Forest	62.7%	37.3%	none
Simplcart	66.7%	33.3%	none

Table 6.10: Comparative analysis for positive bike prediction

	Precision	Recall	F-measure	ROC Area
Proposed Algorithm	0.656	0.108	0.185	0.625
C4.8(J48)	0.383	0.372	0.377	0.684
Random Tree	0.374	0.341	0.357	0.676
Random Forest	0.470	0.303	0.369	0.749
Simplcart	0.655	0.109	0.187	0.626

Table 6.11: Comparative analysis for negative bike prediction

	Precision	Recall	F-measure	ROC Area
Proposed Algorithm	0.910	0.994	0.950	0.625
C4.8(J48)	0.931	0.934	0.933	0.685
Random Tree	0.928	0.937	0.933	0.676
Random Forest	0.926	0.962	0.944	0.749
Simplcart	0.920	0.992	0.950	0.626

Table 6.12: Comparative analysis for Absence of Heart Disease prediction

	Precision	Recall	F-measure	ROC Area
Proposed Algorithm	0.796	0.86	0.827	0.786
C4.8(J48)	0.796	0.86	0.827	0.786
Random Tree	0.74	0.74	0.74	0.703
Random Forest	0.79	0.827	0.808	0.872
Simplcart	0.788	0.82	0.804	0.791

Table 6.13: Comparative analysis for Presence of Heart Disease prediction

	Precision	Recall	F-measure	ROC Area
Proposed Algorithm	0.806	0.725	0.763	0.786
C4.8(J48)	0.806	0.725	0.763	0.786
Random Tree	0.675	0.675	0.675	0.703
Random Forest	0.77	0.725	0.747	0.872
Simplcart	0.763	0.763	0.744	0.791

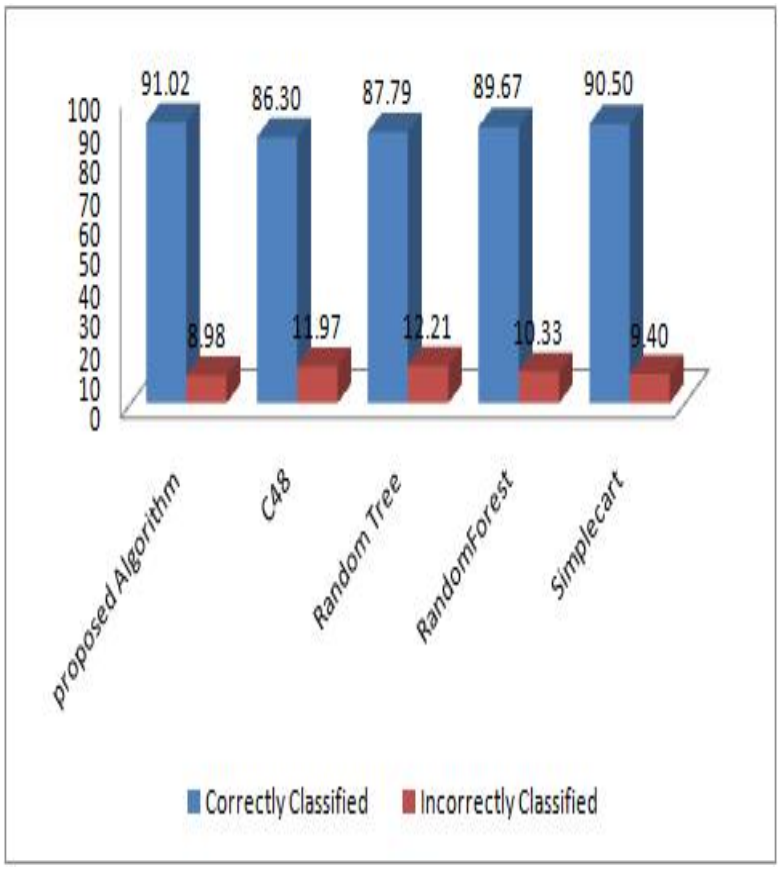

Figure 6.3: Bike Buyer Class Prediction

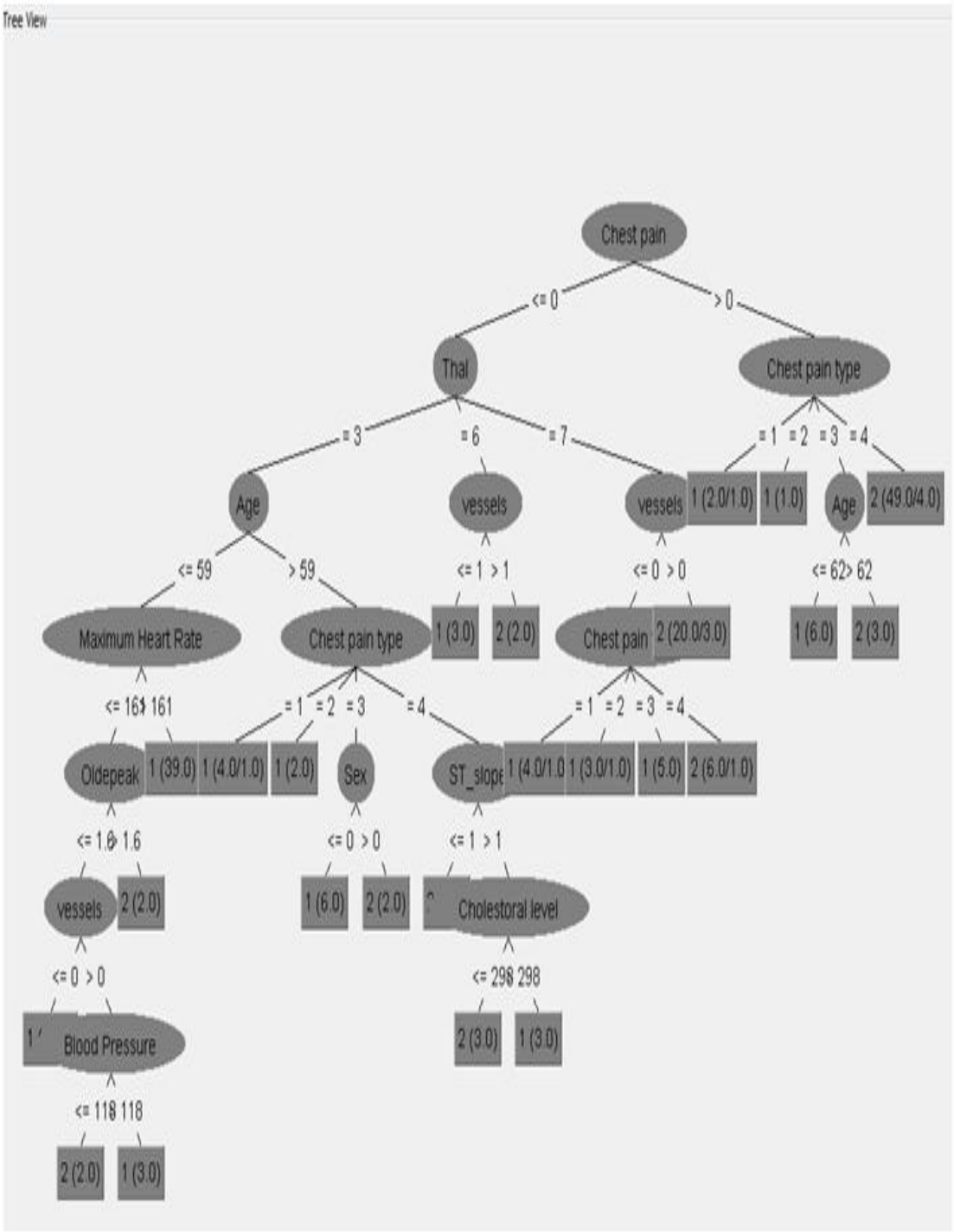

Figure 6.4: Heart Disease Prediction Decision Tree

Chapter 7

Discussion and Conclusions

The primary objective of this dissertation is to improve the predictive ability of the decision tree algorithm in order to determine the possibility that a customer will churn, predict future churn rates, and establish customer relationship management techniques to achieve customer loyalty. Customer churn prediction with decision tree algorithms is not always accurate especially when a data set is skewed, large and distinct; hence, an enhancement to the decision tree algorithms is needed to ensure good churn prediction. Chapter 6 discusses how the decision tree algorithm was enhanced and implemented in order to arrive at good customer churn prediction.

The main limitation in implementing the proposed algorithm is in obtaining current transaction data from organizations. Most establishments find it difficult to release their transaction data because of privacy concerns. There is also the risk that competitors in the same line of business may have access to the data if released to the public. But I am grateful to Mi-

crosoft [80], De Paul University, Chicago, USA, data resource [84], and V.A. Medical Center, Long Beach, and the Cleveland Clinic Foundation [26] for providing the data needed for implementation of the proposed algorithm. Other limitations encountered during the research process include the following:

- Customer Churning is generally a low rate event. As a result, it is difficult to identify customer movement from transaction data especially when the rate of change of data items is very small.

- There are the issues of choosing between test-set data validation and K-fold (10-fold) cross data validation. In test-data validation, about 30% of the data set is used in validating the model, while the remaining 70% is used as training data set. Test-set model validation is easy to implement and gives an accurate result when there is sufficient data in order to avoid overfitting when small training size is used.

  Ten fold data set validation uses about 10% of the data set in validating the model; it gives more accurate result compared to test-set data validation especially when the data set is very small, as all the data records are used for both training and validation. There is one validation for each record (Ron, 1995) [107].

- A decision has to be taken between estimating missing attribute values and deleting records with missing attribute values. In estimating the missing attribute values, local estimation techniques, are preferred over global estimation techniques because in local estimation techniques the missing attribute value is estimated based on the most proba-

ble value within the same class label; this missing attribute value is compared to global estimation that is done without reference to the class label [144].

- Difficulty appears in selecting attributes/features that are of high significance with respect to database instances, in order to obtain good prediction/classification results and reduce classification errors.

This research work focuses on exploring ways of enhancing SPRINT decision tree algorithms to design a good churning model that will predict the probability that a customer will churn. In the business environment, the customer is the "King", and no business entity prospers or survives economic turmoil without having a good customer base and sustainable customer loyalty.

The decision tree algorithm is chosen for enhancement from among other classification/ prediction algorithms because of the following features:

- Decision tree is an eager learning algorithm and produces models that are easy to interpret.
- It handles both categorical and continuous attributes.
- It handles noise and outliers in data sets efficiently.
- It handles missing attributes efficiently.

- Decision tree algorithms do not necessarily require additional information when building the model apart from what is available in the training data set used in building the model [105].

- It is based on rules that are easily integrated to the database (SQL back-engine).

7.1 Main Contributions

Section 1.2 of this dissertation provides the main contributions of the research work to the science and business community. A number of experiments using the proposed Enhanced decision tree algorithm, as well as the system that identifies a good cluster validation algorithm, were performed. We conducted a literature survey on cluster validation algorithms and commonly used decision tree algorithms. We published and presented four papers based on our research work submitting to an international conference and to a journal: the International Journal of Computer Science and Security (IJCSS) and the International Conference on Artificial Intelligence and Pattern Recognition (AIPR-09); all papers were peer reviewed. A summary of our work is stated as follows:

Customer Churn prediction is used to determine accurately customers who are in the risky group of defecting to a competitor's business establishment. Thus, a prediction tool that is fast and scalable with the ability to interpolate with a database back engine which explores the customer database constructively is needed. Most decision tree algorithms available in literature fall short of the features of a good churn prediction tool. The proposed algorithm

incorporates all the good features of SPRINT decision tree algorithms. It uses gain ratio impurity measure instead of gini index to determine the best splitting point on a decision tree node. The splitting attribute approach takes care of skewed data and data that are large and distinct. It equally clusters the customer data record before applying the proposed algorithm to the records. A comparative analysis of the proposed algorithm and other decision tree algorithms were conducted experimentally. The analysis was based on customer churn prediction using the retail and financial and health data sets. A classification accuracy above 90% and a good improvement of other classification accuracy measures were obtained. Thus, our algorithm is suitable in churn prediction algorithm as compared to other decision tree algorithms.

7.2 Future Work

Results of the experiments show that the proposed algorithm performed better than other decision tree prediction algorithms with regard to prediction accuracy and in reducing classification errors. Further work that will implement other impurity splitting measures like entropy into our algorithm and comparing it with the results obtained from gain ratio attribute is being considered. We are considering implementing the proposed algorithm using multiple processors in parallel form. The implementation will lead to increase in processing time and prediction accuracy as suggested by Shafer [116] in their implementation of SPRINT decision tree algorithm.

Bibliography

[1] R. Agrawal and Srikant R. Fast algorithms for mining association rules. *VLDB*, pages 487–99, Sep 12-15 1994.

[2] Nikkei AI. Special edition: General review of expert systems in use. *Nikkei AI. Special Issue*, Winter 1991.

[3] M. Anyanwu and S. Shiva. Application of enhanced decision tree algorithm to churn analysis. In *2009 International Conference on Artificial Intelligence and Pattern Recognition (AIPR-09)*, Orlando Florida, 2009.

[4] F. Azuaje. A cluster validity framework for genome expression data. *Bioinformatics*, 18(2):319–320, 2002.

[5] C. Barbaranell. Evaluating cluster analysis solutions: an application to the italian neo personality inventory. *European Journal of Personality*, 16(1):43–55, 2002.

[6] L. B$\breve{a}$dulescu. Data mining algorithms based on decision trees. Annals of the ORADEA university preprint (2009), available at http://www.imtuoradea.ro/auo.fmte/files-2006, n.d.

[7] S. Bergamaschi, F. Guerra, M. Orsini, and C. Sartori. Extracting relevant attribute valuesfor improved search. *IEEE Internet Computing*, 11(5):26–35, Sept/Oct 2007. (special issue on Semantic-Web-Based Knowledge Management).

[8] A. Berson, S. Smith, and K. Thearling. *Building data mining applications for CRM*. NY: McGraw-Hill, New York, 2002.

[9] J. C. Bezdek. Pattern recognition with fuzzy objective function. *IEEE Press, New York*, pages 483–495, 1981.

[10] N. Bolshakova, Z. Anton, and C. P$\acute{a}$draig. Comparison of the data-based and gene ontology-based approaches to cluster validation methods for gene microarrays. In *Proceeding of the 19th IEEE Symposium on Computer-based Medical*, pages 539–543, 2006b.

[11] N. Bolshakova and F. Azuaje. Cluster validation techniques for genome expression data. *Signal processing*, 19:2494–5, 2003a.

[12] N. Bolshakova and F. Azuaje. Machaon cve:cluster validation for gene expression data. *Bioinformatics*, 19:2494–5, 2003b.

[13] N. Bolshakova, F. Azuaje, and P. Cunningham. A knowledge-driven approach to cluster validity assessment. *Bioinformatics*, 21:2546–47, 2005.

[14] N. Bolshakova, F. Azuaje, and P. Cunningham. Incorporating biological domain knowledge into cluster validity assessment. In *EvoWorkshops*, pages 13–2, 2006a.

[15] SQL Book. An introduction to data warehouses and data warehousing. Retrieved date: July 07, 2009, Available at http://www.sqlbook.com/Data-Warehousing/Introduction-to-Data-Warehouses-2.aspx, n.d.

[16] O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. *Advanced Lectures on Machine Learning*, pages 169–207, 2004.

[17] L. Breiman, J. Friedman, L. Olshen, and J. Stone. *Classification and Regression trees*. Wadsworth, Belmont, 1984.

[18] S. Centhala. Comparison of bill inmon and ralph kimball paradigm. Retrieved date: July 07, 2009, Available at http://srinicenthala.blogspot.com/2009/01/comparison-of-bill-inmon-and-ralph.html, January 29, 2009.

[19] Y. Chen, K. D. Reilly, A. P. Sprague, and Z. Guan. Seqoptics: A protein sequence clustering method. computer and computational sciences. *IMSCCS '06'*, 1:69–75, 2006.

[20] C. Chou, M. C. Su, and E. Lai. new cluster validity measure for clusters with different densities. In *Iasted international conference on intelligent systems and control*, pages 276–281, 2003.

[21] Y. Chunmei, S. Wan, and G. Xiaofeng. Effectivity of internal validation techniques for gene clustering. *Springer Berlin / Heidelberg*, 2006.

[22] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw-Hill, New York, NY, 2008.

[23] T. Davenport and L. Prusak. Working knowledge. *Harvard Business School Press*, 1998.

[24] E. Davide. Methods for intelligent systems. Retrieved date: November 2009, Available at http://home.dei.polimi.it/eynard/200802msi/handout-lecture-e5.pdf, 2008.

[25] J. L. Davies and D. W. Boudlin. A cluster separation measure. In *IEE Transactions on Pattern Analysis and Machine Intelligence*, 1979.

[26] R. Detrano, A. Janosi, W. Steinbrum, M. Pfisterer, J. Schmid, S. Sandhu, K. Guppy, S. Lee, and V. Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology*, 64:304–310, 1989.

[27] M. J. Druzdzel and R. R. Flynn. Decision support systems. Retrieved date: July 27, 2009, Available at http://www.sis.pitt.edu/dsl, n.d.

[28] W. Du and Z. Zhan. Building decision tree classifier on private data. In *Proceedings of the IEEE international conference on Privacy, security and data mining*, pages 1–8, Maebashi City, Japan, 2002.

[29] J. C. Dunn. Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4:95–104, 1974.

[30] R. Falbo, D. Arantes, and A. Natali. Integrating knowledge management and groupware in a software development enviroment. In *Proceeding of the Fifth International Conference on Practical Aspects of Knwoledgement, PAKM*, Vienna, Austria, December 2004.

[31] U. Fayyad, G. Piatetsky-shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.

[32] J. B. Ferreira, M. Vellasco, M. A. Pacheco, and C. H. Barbosa. Data mining techniques on the evaluation of wireless churn. In *In ESANN*, pages 483–488, 2004.

[33] P. Filzmoser and M. Templ. Cluster analysis with application to data from geochemistry. Vienna University of Technology, Wiedner Hauptstr. 8-10, A-1040 Vienna, Austria, 2003.

[34] J. M. Firestone. Data warehouses, data marts, and data warehousing. Retrieved date: July 07, 2009, Available at http://www.tgc.com/dsstar/01/0320/102809.html, n.d.

[35] S. French and M. Turoff. Decision support systems. *Communications of the ACM*, 50(3):39–40, 2007.

[36] M. Garofalakis, D. Hyun, R. Rastogi, and K. Shim. Efficient algorithms for constructing decision trees with constraints. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 335–339, New York, USA, 2000.

[37] J. Gehrke, R. Ramakrishnan, and V. Ganti. Rainforest - a framework for fast decision tree construction of large datasets. In *Proceedings of the 24th VLDB conference*, pages 416–427, New York, USA, 1998.

[38] Gepas. Cluster accuracy analysis tool caat. preprint (2005), available at http://gepas3.bioinfo.cipf.es/cgi-bin/tutoX?c=CAAT/caat.config, 2005.

[39] R. M. Goodman and P. Smyth. Decision tree design from a communication theory standpoint. *IEEE Transactions on Information Theory*, 34(5):979–994, Sep 1988.

[40] S. Gupta, D. Hanssens, B. Hardie, W. Kahn, V. Kumar, N. Lin, N. Ravishanker, and S. Sriram. Modeling customer lifetime value. *Journal of Service Research*, 9(2):139–155, 2006.

[41] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145, 2001.

[42] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering validity methods: Part: I. *ACM press New York*, 31(2), 2002a.

[43] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering validity methods: Part: Ii. *ACM press New York*, 31(3), 2002b.

[44] H. Hamilton, E. Gurak, L. Findlater, and W. Olive. Overview of decision trees. Retrieved date: January 3, 2009, Available at http://www.cs.uregina.ca/dbd/cs831/notes/ml/dtrees/, 2001.

[45] J. Handi, J. Knowles, and D. Kell. Computational cluster validation post-genomic data analysis. *Bioinformatics Review Journal*, 21(15):3201–3212, 2005.

[46] C. W. Holsapple. Knowledge management support of decision making. *Decision Support Systems*, 31(3):1–3, May 2001.

[47] S. J. Hong and S. M. Weiss. Advances in predictive model generation for data mining. In *Proceedings 1st International Workshop Machine Learning and Data Mining in Pattern Recognition*, page 12, 1999.

[48] D. W. Huang, Z.and Cheung and M. K. Ng. An empirical study on the visual cluster validation method with fastmap. *Database systems for advanced applications*, pages 84–91, 2001.

[49] L. Hubert and J. Schultz. Quadratic assignment as a general data-analysis strategy. *British Journal of Mathematical and Statistical Psychologie*, 29:190–241, 1976.

[50] S. H. Hung, D. C. Yen, and H. H. Wang. Applying data mining to telecom churn management. *Expert Systems with Applications*, 31(3):515–524, 2006.

[51] E. B. Hunt, J. Marin, and P. J. Stone. *Experiments in induction*. Academic Press, New York, USA, 1966.

[52] H. Hwang, T. Jung, and E. Suh. An ltv model and customer segmentation based on customer value: a case study on the wireless telecommunication industry. *Expert systems with applications*, 26:181–188, 2004.

[53] L. Hyafil and R.L. Rivest. Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5(1):15–17, 1976.

[54] IBM. Cognos 8 business intelligence. Retrieved date: July 27, 2009, Available at http://www-01.ibm.com/software/data/cognos/products/cognos-8-business-intelligence, n.d.

[55] B. Inmon. Data mart does not equal data warehouse. *InfoManagement Direct*, November 1999.

[56] W. H. Inmon. What is a data warehouse? *Prism Tech Topic*, 1(1), 1995.

[57] A. Jaccard. Nouvelles recherches sur la distribution florale. *Bull Soc Vaudoises Sci Nat*, 44:223270, 1908.

[58] N. Jethwa. Dashboards - business intelligence. Retrieved date: July 07, 2009, Available at http://www.appsbi.com, 2009.

[59] E. Johansson and P. Lindberg. Low back pain patients in primary care: Subgroups based on the multidimensional pain inventory. *International journal of behavioral medicine*, 7(4):340–352, 2000.

[60] J. Kalbfleisch and Ross Prentice. *Statistical Analysis of Failure Time Data*. Wiley, New York, 1980.

[61] T. Kardi. Tutorial on decision tree. Retrieved date: October 12, 2009, Available at http://people.revoledu.com/kardi/tutorial/decisiontree, 2009.

[62] R. Kasturi, J.and Acharya and M. Ramanathan. An information theoretic approach for analyzing temporal patterns of gene expression. *International journal of behavioral medicine*, 19(4):449–458, 2003.

[63] T. M Khoshgoftaar and E. B. Allen. Logistic regression modelling of software quality. *Inter.Journal of Realiabilit,Quality and Safety Engineering*, 6:303–317, 1999.

[64] R. Kimball, L. Reeves, M. Ross, and W. Thornthwaite. *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses with CD Rom*. Number 1. Wiley, John and Sons, Incorporated, January 1998.

[65] R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. Wiley, John and Sons, Incorporated, 2nd edition, 2002.

[66] R. Kufrin. *Decision trees on parallel processors*. Parallel Processing for Artificial Intelligence for Elsevier Science, 1997.

[67] L. Kuncheva and D. P. Vetrov. valuation of stability of k-means cluster ensembles with respect to random initialization. *IEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1798–1808, 2006.

[68] Katherine L. and V. Zeithaml. Return on marketing: Using customer equity to focus marketing strategy. *Journal of Marketing*, 68(1):109–26, 2004.

[69] C. Legány, F. Kovács, and A. Babos. Cluster validity measurement techniques. In *Proceedings of the 5th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, pages 388–393, 2006.

[70] R. J. Lewis. An introduction to classification and regression tree (cart) analysis. In *2000 Annual Meeting of the Society for Academic Emergency Medicine*, Francisco, California, 2000.

[71] X. C. Ling, Q. Yang, J. Wang, and S. Zhang. Decision trees with minimal costs. In *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

[72] J. J. H. Liou. A novel decision rules approach for customer relationship management of the airline market. *Expert Systems with Applications*, 36(3):4374–4381, 2009.

[73] R. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 22, 1987.

[74] T. Liu, Y.and zyer, R. Alhajj, and K. Barker. Integrating multi-objective genetic algorithm and validity analysis for locating and ranking alternative clustering. *Informatica*, 29:33–40, 2005.

[75] R. Loganantharaj, S. Cheepala, and J. Clifford. Metric for measuring the effectiveness of clustering of dna microarray expression. *BMC Bioinformatics*, 2006.

[76] B. Lokken. Business intelligence: An intelligent move or not? Retrieved date: July 07, 2009, Available at http://bi.ittoolbox.com/documents/document.asp?i=1387, 2002.

[77] M. Mehta, R. Agrawal, and J. Rissanen. Mdl-based decision tree pruning. In *International conference on knowledge discovery in databases and data mining(KDD-95)*, Montreal, Canada, 1995.

[78] M. Mehta, R. Agrawal, and J. Rissanen. Sliq: A fast scalable classifier for data mining. In *In EDBT 96*, Avignon, France, 1996.

[79] D. Michie, D. Spiegelhalter, J., and C. C. Taylor. Machine learning, neural and statistical classfication. *Ellis Horword*, 1994.

[80] Microsoft®. Microsoft sql server2008 business intelligence. Retrieved date: July 27, 2009, Available at http://www.microsoft.com/sqlserver/2008/en/us/business-intelligence.aspx, 2008.

[81] MicroStrategy. Microstrategy 9. Retrieved date: July 27, 2009, Available at http://www.microstrategy.com, 2009.

[82] MicroStrategy. The 5 styles of business intelligence: Industrial-strength business intelligence. Retrieved date: July 07, 2009, Available at http://www.microstrategy.com/Download/files/Solutions/5Styles/5Styles.pdf?cid=2119styles, n.d.

[83] E. Minami and T. Hirata. An expert system for large scale fault diagnosis in steel manufacturing. In *Proceeding of World Congress on Expert Systems*, pages 1311–1318, 1991.

[84] B. Mobasher. Data mining with weka. Retrieved date: July 07, 2009, Available at http://www.maya.cs.depaul.edu, 2009.

[85] L. Modistte. Milking wireless churn for profit. *Telecommunications*, 33(3), 1999.

[86] T. Murata. Visualizing the structure of web communities based on data acquired from a search engine. industrial electronics. *IEEE transactions*, 50(5):860–866, 2003.

[87] O.J. Murphy and R.L. McCraw. Designing storage efficient decision trees. *IEEE Transactions of Computers*, 40(3):315–320, 1991.

[88] S. Murthy and S. Salzberg. Decision tree induction: How effective is the greedy heuristic? In *In Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pages 222–227. Morgan Kaufmann, 1995.

[89] T. Mutanen. Customer churn analysis-a case study. Retrieved date: January 27, 2009, Available at http://www.vtt.fi/inf/julkaisut/muut/2006, 2001.

[90] M. Nadeem and S. A. H Jaffri. Application of business intelligence in banks (pakistan). Retrieved date: July 07, 2009, Available at http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0406004, 2004.

[91] Oracle. Oracle business intelligence suiteenterprise edition plus. Retrieved date: July 27, 2009, Available at http://www.oracle.com/appserver/business-intelligence/enterprise-edition.html, n.d.

[92] B. Paige and P. Amy. *Business driven information systems*. The McGraw-Hill companies Incorporated, 2008.

[93] T. Palpanas. Knowledge discovery in data warehouses. *SIGMOD Record*, 2000:88–100, 2000.

[94] W. Peng, J. Chen, and H. Zhou. An implementation of ide3
− decision tree learning algorithm. University of New South Wales, School of Computer Science and Engineering, Sydney, NSW 2032, Retrieved date: May 13, 2009, Available at web.arch.usyd.edu.au/ wpeng/DecisionTree2.pdf, n.d.

[95] P. Pfeifer and R. Carraway. Modeling customer relationships as markov chains. *Journal of Interactive Marketing*, 14(2):43–55, 2000.

[96] J. Pješivac-Grbović, T. Angskun, G. Bosilca, G. E. Fagg, and J. J. Dongarra. Decision trees and mpi collective algorithm selection problem. tech. rep. ut-cs-06-586. The University of Tennessee at Knoxville, Computer Science Department, Available at http://www.cs.utk.edu/ library/2006.html, 2006.

[97] V. Podgorelec, P. Kokol, B. Stiglic, and I. Rozman. Decision trees: an overview and their use in medicine. *Journal of Medical Systems*, 26(5):445–463, 2002.

[98] D. Power. Types of decision support systems (dss). Retrieved date: July 27, 2009, Available at http://www.gdrc.org/decision/dss-types.html, n.d.

[99] P. Putten and M. Someren. Coil challenge 2000: The insurance company case. *Sentient Machine Research and Leiden Institute of Advanced Computer Science Technical Report*, 2000.

[100] J. R. Quinlan. *Learning efficient classification procedures and their application to chess end games*, volume 1. Morgan Kaufmann, San Mateo, CA, 1983.

[101] J. R. Quinlan. Induction of decision trees. *Machine Leaning*, 1:81–106, 1986.

[102] J. R. Quinlan. Simplifying decision trees. *International Journal of Manmachine Studies*, pages 221–234, 1987.

[103] J. R. Quinlan. *C45: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[104] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[105] R. Rastogi and K. Shim. Public: A decision tree classifier that integrates building and pruning. In *Proceedings of the 24th VLDB Conference*, pages 404–415, New York, 1998.

[106] M. Richeldi and A. Perrucci. Churn analysis case study. Telecom Italia Lab, Torino, Italy, 2002.

[107] K. Ron. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1137–1143, 1995. Morgan Kaufmann, San Mateo, Available at http://citeseer.ist.psu.edu/kohavi95study.html.

[108] S. Rosset, E. Neumann, U. Eick, N. Vatnik, and I. Idan. Evaluation of prediction models for marketing campaigns. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 456–461, 2001.

[109] P. J. Rousseuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational Applications in Mat*, 20:53–65, 1987.

[110] T. Rust, R., T. Ambler, G. Carpenter, V. Kumar, and R. Srivastava. Measuring marketing productivity: current knowledge and future directions. *Journal of Marketing*, 68:76–89, 2004.

[111] A. Sachenko. Business support systems. Retrieved date: July 27, 2009, Available at http://www.scribd.com/doc/396842/Business-Support-Systems, 2007.

[112] G. S. Sadesky. Cluster analysis and its application in standard setting. University of Alberta, n.d.

[113] G. K. Saha. Business intelligence computing issues. *ACM*, 8(25), 2007.

[114] SAP. Sap businessobjects intelligence platform. Retrieved date: July 27, 2009, Available at http://www.sap.com/solutions/sapbusinessobjects/large/intelligenceplatform/index.epx, n.d.

[115] K. Sattler and O. Dunemann. Sql database primitives for decision tree classifiers. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 379–386, 2001.

[116] J. Shafer, R. Agrawal, and M. Mehta. Sprint: A scalable parallel classifier for data mining. In *Proceeding of the 22nd international conference on very large data base*, Mumbai(Bombay), India, 1996.

[117] M. J. Shawa, C. Subramaniama, G. W. Tana, and M. E. Welgeb. Knowledge management and data mining for marketing. *Decision Support Systems*, 31(1):127–137, May 2001.

[118] J. P. Shim, M. Warkentin, J. F. Courtney, D. J. Power, R. Sharda, and C. Carlsson. Past, present and future of decision support technology. *Decision Support Systems*, 33(2):111–126, 2002.

[119] S. G. Shiva, S. B. Lee, L. A. Shala, and C. B. Simmons. Knowledge management in global software development. In *International Symposium On Advances in Computer and Sensor Networks and Systems*, pages 644–650, Zhengzhou, China, April 7-11 2008.

[120] W. Silvert. Modeling as a discipline. *Int. J. General Systems*, 30(8):26, 2001.

[121] J. Simpson and E. Weiner. *Oxford English Dictionary* . Oxford University Press, October 2009.

[122] Rosella Software. Customer retention marketing- software analytics, and methods. Retrieved date: January 27, 2009, Available at http://www.roselladb.com/customer-retention.htm, 2001.

[123] R. H. Sprague and H. J Watson. *Decision Support for Management*. Prentice-Hall, 1996.

[124] A. Srivastava and V. Singh. Methods to reduce i/o for decision tree classifiers. IBM T.J Watson Reseach center, n. d.

[125] A. Srivastava, V. Singh, E. Han, and V . Kumar. An efficient, scalable, parallel classifier for data mining. University of Minnesota, Computer Science, technical report, USA., 1997.

[126] A. Srivastava, V. Singh, E. Han, and V. Kumar. Parallel formulations of decision-tree classification algorithms. *Data Mining and Knowledge Discovery, an international journa*, pages 237–261, 1998.

[127] S. Stein, M. Eissen, and F. Wibrock. On cluster validity and the information need of users. *Artificial intelligence and applications*, 403:203, 2003.

[128] J. Sudhakar and S. Rajagopalan. An information theory approach for validating clusters in microarray data. In *12th International Conference on Intelligent Systems for Molecular Biology and 3rd European Conference on Computational Biology Glasgow Uk*, 2004.

[129] P. Sulikowski. Mobile operator customer classification in churn analysis. In *SAS Global Forum Conference*, 2008.

[130] P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison Wesley, New York, 2006.

[131] techfAQ. What is a decision support system? Retrieved date: July 07, 2009, Available at http://www.tech-faq.com/decision-support-system.shtml, n.d.

[132] S. Theodoridis and K. Koutroubas. *Pattern recognition*. Academic Press, 1999.

[133] C. Todman. *Designing a data warehouse: supporting customer relationship management*. Prentice Hall PTR, 2001.

[134] Turban, Leidner, McLea, and Wetherbe. *Information Technology for Management: Transforming Organizations in the Digital Economy* . Wiley, 6th edition, 2008.

[135] P. Utgoff and C. Brodley. An incremental method for finding multivariate splits for decision trees, machine learning. In *Proceedings of the Seventh International Conference*, page 58, 1990.

[136] P. Van Der, J. Puttan, N. Kok, and A. Gupta. Data fusion through statistical matching. preprint (2002), available at http://papers.ssrn.com/abstract=297501, 2002.

[137] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[138] M. Vazirgiannis, M. Halkidi, and D. Gunopulos. Uncertainty handling and quality assessment in data mining. *Springer*, 2003.

[139] Fan W., H. Wang, P. Yu, and S. Ma. Is random model better? on its accuracy and efficiency. In *Proceedings of the 3rd IEEE international conference on data mining*, 2003.

[140] J. Wang and J. Chiang. A cluster validity measure with a hybrid parameter search method for the support vector clustering algorithm. *The Journal of the Pattern Recognition society*, 41(2):506–520, 2008.

[141] Y. F. Wang, D. Chiang, C. J. Lin, and I. L. Lin. A recommender system to avoid customer churn: A case study. *Expert Systems with Applications*, 2008. doi:10.1016/j.eswa.2008.10.089.

[142] S. Watanabe, T. Ito, T. Ozono, and T. Shintani. A paper recommendation mechanism for the research support system papits. *Data engineering issues in E-Commerce*, 41:71–80, 2005.

[143] X. Wen, G. Hu, and X. Yang. Cbers-02 remote sensing data mining using decision tree algorithm. In *First International Workshop on Knowledge Discovery and Data Mining*, pages 86–89, 2008.

[144] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[145] Y. Xie, V. Raghavan, and X. Zhao. 3m algorithm: finding an optimal fuzzy custer scheme for proximity data. University of Louisiana, 2000.

[146] M. Xu, J. Wang, and T. Chen. Improved decision tree algorithm: $id3^{+}$, intelligent computing in signal processing and pattern recognition. *Intelligent Computing in Signal Processing and Pattern Recognition*, 345:141–149, 2006.

[147] L. Yan, R. H. Wolniewicz, and R. Dodier. Predicting customer behavior in telecommunications. *Intelligent Systems, IEEE*, 19:50–58, 2004.

[148] C. Yang, E. Zeng, T. Li, and G. A Narasimhan. knowledge-driven method to evaluate multi-source clustering. In *1$Congrs$ Parallel and distributed processing and applications ISPA 2005 Workshops*, 2005.

[149] K. Y. Yeung, D. R Haynor, and W. L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17:309–18, 2001.

[150] H. Zhao, J. Liang, and H. Hu. Clustering validity based on the improved hubert γ statistic and the separation of clusters. In *Proceedings of the first international conference on innovative computing, information and control*, volume 2, pages 539–543, 2006.

[151] J. Zimmermann, Z. Liptak, and S. Hazelhurst. A method for evaluating the quality of string dissimilarity measures and clustering algorithms for est clustering. *Bioinformatics and bioengineering*, pages 301–309, 2004.

Appendix

```
package weka.classifiers;

import weka.core.Attribute;

import weka.core.Capabilities;

import weka.core.Capabilities.Capability;

import weka.core.Instance;

import weka.core.Instances;

import weka.classifiers.Classifier;

public class WekaWrapper

extends Classifier {

/**

* Returns only the toString() method.

*

* @return a string describing the classifier

*/

public String globalInfo() {

return toString();

}

/**

* Returns the capabilities of this classifier.

*
```

```
* @return the capabilities
*/
public Capabilities getCapabilities() {
weka.core.Capabilities result = new weka.core.Capabilities(this);
result.enable(weka.core.Capabilities.Capability.NOMINAL_ATTRIBUTES);
result.enable(weka.core.Capabilities.Capability.BINARY_ATTRIBUTES);
result.enable(weka.core.Capabilities.Capability.UNARY_ATTRIBUTES);
result.enable(weka.core.Capabilities.Capability.EMPTY_NOMINAL_ATTRIBUTES);
result.enable(weka.core.Capabilities.Capability.NUMERIC_ATTRIBUTES);
result.enable(weka.core.Capabilities.Capability.DATE_ATTRIBUTES);
result.enable(weka.core.Capabilities.Capability.MISSING_VALUES);
result.enable(weka.core.Capabilities.Capability.NOMINAL_CLASS);
result.enable(weka.core.Capabilities.Capability.BINARY_CLASS);
result.enable(weka.core.Capabilities.Capability.MISSING_CLASS_VALUES);
result.setMinimumNumberInstances(0);
return result;
}
/**
* only checks the data against its capabilities.
*
* @param i the training data
```

```
*/
public void buildClassifier(Instances i) throws Exception {
// can classifier handle the data?
getCapabilities().testWithFail(i);
}
/**
* Classifies the given instance.
*
* @param i the instance to classify
* @return the classification result
*/
public double classifyInstance(Instance i) throws Exception {
Object[] s = new Object[i.numAttributes()];
for (int j = 0; j < s.length; j++) {
if (!i.isMissing(j)) {
if (i.attribute(j).isNominal())
s[j] = new String(i.stringValue(j));
else if (i.attribute(j).isNumeric())
s[j] = new Double(i.value(j));
}
}
```

```
// set class value to missing
s[i.classIndex()] = null;
return WekaClassifier.classify(s);
}
/**
* Returns only the classnames and what classifier it is based on.
*
* @return a short description
*/
public String toString() {
return ”Auto-generated classifier wrapper, based on weka.classifiers.trees.Sprint (generated
with Weka 3.5.8).\n” + this.getClass().getName() + ”/WekaClassifier”;
}
/**
* Runs the classfier from commandline.
*
* @param args the commandline arguments
*/
public static void main(String args[]) {
runClassifier(new WekaWrapper(), args);
}
```

```
}
class WekaClassifier {
public static double classify(Object[] i)
throws Exception {
double p = Double.NaN;
p = WekaClassifier.N4efaa9580(i);
return p;
}
static double N4efaa9580(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
p = 1;
} else if (((Double) i[8]).doubleValue() <= 1.0) {
p = WekaClassifier.N2a83f8ea1(i);
} else if (((Double) i[8]).doubleValue() > 1.0) {
p = WekaClassifier.N21fd5f1f81(i);
}
return p;
}
static double N2a83f8ea1(Object []i) {
double p = Double.NaN;
```

```
if (i[11] == null) {
p = 1;
} else if (((Double) i[11]).doubleValue() <= 66.0) {
p = WekaClassifier.N3e4ae0742(i);
} else if (((Double) i[11]).doubleValue() > 66.0) {
p = 1;
}
return p;
}
static double N3e4ae0742(Object []i) {
double p = Double.NaN;
if (i[4] == null) {
p = 1;
} else if (((Double) i[4]).doubleValue() <= 2.0) {
p = WekaClassifier.N25d5ff9f3(i);
} else if (((Double) i[4]).doubleValue() > 2.0) {
p = WekaClassifier.N1ece6ba176(i);
}
return p;
}
static double N25d5ff9f3(Object []i) {
```

```
double p = Double.NaN;
if (i[11] == null) {
p = 1;
} else if (((Double) i[11]).doubleValue() <= 33.0) {
p = WekaClassifier.N48133214(i);
} else if (((Double) i[11]).doubleValue() > 33.0) {
p = WekaClassifier.N3d3ab25011(i);
}
return p;
}
static double N48133214(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
p = 1;
} else if (((Double) i[8]).doubleValue() <= 0.0) {
p = WekaClassifier.N3e43bdd65(i);
} else if (((Double) i[8]).doubleValue() > 0.0) {
p = WekaClassifier.N34dff0d69(i);
}
return p;
}
```

```
static double N3e43bdd65(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 0;
} else if (((Double) i[3]).doubleValue() <= 30000.0) {
p = WekaClassifier.N51644c896(i);
} else if (((Double) i[3]).doubleValue() > 30000.0) {
p = WekaClassifier.Nfdce01e7(i);
}
return p;
}
static double N51644c896(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 1;
} else if (((Double) i[0]).doubleValue() <= 13054.0) {
p = 1;
} else if (((Double) i[0]).doubleValue() > 13054.0) {
p = 0;
}
return p;
```

```
}
static double Nfdce01e7(Object []i) {
double p = Double.NaN;
if (i[10] == null) {
p = 1;
} else if (i[10].equals(”Europe”)) {
p = WekaClassifier.N6dfdd9b48(i);
} else if (i[10].equals(”Pacific”)) {
p = 1;
} else if (i[10].equals(”North America”)) {
p = 1;
}
return p;
}
static double N6dfdd9b48(Object []i) {
double p = Double.NaN;
if (i[9] == null) {
p = 1;
} else if (i[9].equals(”0-1 Miles”)) {
p = 1;
} else if (i[9].equals(”10+ Miles”)) {
```

```
p = 1;
} else if (i[9].equals("5-10 Miles")) {
p = 1;
} else if (i[9].equals("2-5 Miles")) {
p = 1;
} else if (i[9].equals("1-2 Miles")) {
p = 0;
}
return p;
}
static double N34dff0d69(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 1;
} else if (((Double) i[3]).doubleValue() <= 10000.0) {
p = WekaClassifier.N5a3c8cb810(i);
} else if (((Double) i[3]).doubleValue() > 10000.0) {
p = 1;
}
return p;
}
```

```
static double N5a3c8cb810(Object []i) {
double p = Double.NaN;
if (i[1] == null) {
p = 1;
} else if (i[1].equals("Married")) {
p = 1;
} else if (i[1].equals("Single")) {
p = 0;
}
return p;
}
static double N3d3ab25011(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 1;
} else if (((Double) i[3]).doubleValue() <= 20000.0) {
p = 1;
} else if (((Double) i[3]).doubleValue() > 20000.0) {
p = WekaClassifier.N5890c19712(i);
}
return p;
```

```
}
static double N5890c19712(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 1;
} else if (((Double) i[0]).doubleValue() <= 15550.0) {
p = WekaClassifier.N795cdee13(i);
} else if (((Double) i[0]).doubleValue() > 15550.0) {
p = WekaClassifier.N102aeca226(i);
}
return p;
}
static double N795cdee13(Object []i) {
double p = Double.NaN;
if (i[10] == null) {
p = 1;
} else if (i[10].equals("Europe")) {
p = WekaClassifier.N7c02c82514(i);
} else if (i[10].equals("Pacific")) {
p = WekaClassifier.N18a11ec820(i);
} else if (i[10].equals("North America")) {
```

```
p = WekaClassifier.N34d5a63a23(i);
}
return p;
}
static double N7c02c82514(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 1;
} else if (((Double) i[11]).doubleValue() <= 38.0) {
p = WekaClassifier.N1aa234be15(i);
} else if (((Double) i[11]).doubleValue() > 38.0) {
p = 1;
}
return p;
}
static double N1aa234be15(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
p = 1;
} else if (((Double) i[8]).doubleValue() <= 0.0) {
p = WekaClassifier.Nf34b39b16(i);
```

```
} else if (((Double) i[8]).doubleValue() > 0.0) {
p = 1;
}
return p;
}
static double Nf34b39b16(Object []i) {
double p = Double.NaN;
if (i[5] == null) {
p = 0;
} else if (i[5].equals("Graduate Degree")) {
p = WekaClassifier.N4cfb8ce917(i);
} else if (i[5].equals("Partial College")) {
p = WekaClassifier.N16e03c7b18(i);
} else if (i[5].equals("High School")) {
p = 1;
} else if (i[5].equals("Bachelors")) {
p = WekaClassifier.N5ba2baf819(i);
} else if (i[5].equals("Partial High School")) {
p = 1;
}
return p;
```

```
}
static double N4cfb8ce917(Object []i) {
double p = Double.NaN;
if (i[7] == null) {
p = 0;
} else if (i[7].equals("Yes")) {
p = 0;
} else if (i[7].equals("No")) {
p = 1;
}
return p;
}
static double N16e03c7b18(Object []i) {
double p = Double.NaN;
if (i[9] == null) {
p = 1;
} else if (i[9].equals("0-1 Miles")) {
p = 1;
} else if (i[9].equals("10+ Miles")) {
p = 1;
} else if (i[9].equals("5-10 Miles")) {
```

```
p = 1;
} else if (i[9].equals("2-5 Miles")) {
p = 1;
} else if (i[9].equals("1-2 Miles")) {
p = 0;
}
return p;
}
static double N5ba2baf819(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 0;
} else if (((Double) i[11]).doubleValue() <= 37.0) {
p = 0;
} else if (((Double) i[11]).doubleValue() > 37.0) {
p = 1;
}
return p;
}
static double N18a11ec820(Object []i) {
double p = Double.NaN;
```

```
if (i[11] == null) {
p = 1;
} else if (((Double) i[11]).doubleValue() <= 47.0) {
p = WekaClassifier.N79ebe91921(i);
} else if (((Double) i[11]).doubleValue() > 47.0) {
p = 1;
}
return p;
}
static double N79ebe91921(Object []i) {
double p = Double.NaN;
if (i[2] == null) {
p = 0;
} else if (i[2].equals("Female")) {
p = 0;
} else if (i[2].equals("Male")) {
p = WekaClassifier.N6f14a6d722(i);
}
return p;
}
static double N6f14a6d722(Object []i) {
```

```
double p = Double.NaN;
if (i[9] == null) {
p = 0;
} else if (i[9].equals("0-1 Miles")) {
p = 0;
} else if (i[9].equals("10+ Miles")) {
p = 1;
} else if (i[9].equals("5-10 Miles")) {
p = 1;
} else if (i[9].equals("2-5 Miles")) {
p = 1;
} else if (i[9].equals("1-2 Miles")) {
p = 1;
}
return p;
}
static double N34d5a63a23(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 1;
} else if (((Double) i[3]).doubleValue() <= 70000.0) {
```

```
p = 1;
} else if (((Double) i[3]).doubleValue() > 70000.0) {
p = WekaClassifier.N369444ef24(i);
}
return p;
}
static double N369444ef24(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 0;
} else if (((Double) i[11]).doubleValue() <= 38.0) {
p = WekaClassifier.N4039c2e725(i);
} else if (((Double) i[11]).doubleValue() > 38.0) {
p = 1;
}
return p;
}
static double N4039c2e725(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 1;
```

```
} else if (((Double) i[0]).doubleValue() <= 13172.0) {
p = 1;
} else if (((Double) i[0]).doubleValue() > 13172.0) {
p = 0;
}
return p;
}
static double N102aeca226(Object []i) {
double p = Double.NaN;
if (i[9] == null) {
p = 1;
} else if (i[9].equals("0-1 Miles")) {
p = WekaClassifier.N721587f827(i);
} else if (i[9].equals("10+ Miles")) {
p = 1;
} else if (i[9].equals("5-10 Miles")) {
p = WekaClassifier.N54836cd548(i);
} else if (i[9].equals("2-5 Miles")) {
p = WekaClassifier.N2b8d96c450(i);
} else if (i[9].equals("1-2 Miles")) {
p = WekaClassifier.N6e149fc270(i);
```

```
}
return p;
}
static double N721587f827(Object []i) {
double p = Double.NaN;
if (i[10] == null) {
p = 0;
} else if (i[10].equals(”Europe”)) {
p = WekaClassifier.N6391e54b28(i);
} else if (i[10].equals(”Pacific”)) {
p = WekaClassifier.N5f663b7f33(i);
} else if (i[10].equals(”North America”)) {
p = WekaClassifier.N741bf96c37(i);
}
return p;
}
static double N6391e54b28(Object []i) {
double p = Double.NaN;
if (i[4] == null) {
p = 0;
} else if (((Double) i[4]).doubleValue() <= 1.0) {
```

```
p = WekaClassifier.N7f64d1fe29(i);
} else if (((Double) i[4]).doubleValue() > 1.0) {
p = 1;
}
return p;
}
static double N7f64d1fe29(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 0;
} else if (((Double) i[11]).doubleValue() <= 63.0) {
p = WekaClassifier.N3416bc0930(i);
} else if (((Double) i[11]).doubleValue() > 63.0) {
p = 1;
}
return p;
}
static double N3416bc0930(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
p = 0;
```

```
} else if (((Double) i[8]).doubleValue() <= 0.0) {
p = WekaClassifier.N40930d9631(i);
} else if (((Double) i[8]).doubleValue() > 0.0) {
p = 1;
}
return p;
}
static double N40930d9631(Object []i) {
double p = Double.NaN;
if (i[6] == null) {
p = 0;
} else if (i[6].equals("Clerical")) {
p = 0;
} else if (i[6].equals("Professional")) {
p = 0;
} else if (i[6].equals("Manual")) {
p = 0;
} else if (i[6].equals("Skilled Manual")) {
p = WekaClassifier.N766113f832(i);
} else if (i[6].equals("Management")) {
p = 0;
```

```
}
return p;
}
static double N766113f832(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 1;
} else if (((Double) i[0]).doubleValue() <= 19492.0) {
p = 1;
} else if (((Double) i[0]).doubleValue() > 19492.0) {
p = 0;
}
return p;
}
static double N5f663b7f33(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
p = 0;
} else if (((Double) i[8]).doubleValue() <= 0.0) {
p = 0;
} else if (((Double) i[8]).doubleValue() > 0.0) {
```

```
p = WekaClassifier.N30cc3fc734(i);
}
return p;
}
static double N30cc3fc734(Object []i) {
double p = Double.NaN;
if (i[1] == null) {
p = 1;
} else if (i[1].equals("Married")) {
p = WekaClassifier.N316f3ad835(i);
} else if (i[1].equals("Single")) {
p = WekaClassifier.N7d80544236(i);
}
return p;
}
static double N316f3ad835(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 0;
} else if (((Double) i[3]).doubleValue() <= 60000.0) {
p = 0;
```

```
} else if (((Double) i[3]).doubleValue() > 60000.0) {
p = 1;
}
return p;
}
static double N7d80544236(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 1;
} else if (((Double) i[0]).doubleValue() <= 17776.0) {
p = 1;
} else if (((Double) i[0]).doubleValue() > 17776.0) {
p = 0;
}
return p;
}
static double N741bf96c37(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 1;
} else if (((Double) i[3]).doubleValue() <= 50000.0) {
```

```
p = WekaClassifier.N58945d0138(i);
} else if (((Double) i[3]).doubleValue() > 50000.0) {
p = WekaClassifier.N5b3a946640(i);
}
return p;
}
static double N58945d0138(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 1;
} else if (((Double) i[11]).doubleValue() <= 49.0) {
p = 1;
} else if (((Double) i[11]).doubleValue() > 49.0) {
p = WekaClassifier.N74661a1539(i);
}
return p;
}
static double N74661a1539(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 1;
```

```
} else if (((Double) i[11]).doubleValue() <= 50.0) {
p = 1;
} else if (((Double) i[11]).doubleValue() > 50.0) {
p = 0;
}
return p;
}
static double N5b3a946640(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
p = 1;
} else if (((Double) i[8]).doubleValue() <= 0.0) {
p = WekaClassifier.N62e4e35441(i);
} else if (((Double) i[8]).doubleValue() > 0.0) {
p = WekaClassifier.N6886b57b44(i);
}
return p;
}
static double N62e4e35441(Object []i) {
double p = Double.NaN;
if (i[6] == null) {
```

```
p = 1;
} else if (i[6].equals("Clerical")) {
p = 1;
} else if (i[6].equals("Professional")) {
p = WekaClassifier.N2108d28542(i);
} else if (i[6].equals("Manual")) {
p = 1;
} else if (i[6].equals("Skilled Manual")) {
p = 1;
} else if (i[6].equals("Management")) {
p = 1;
}
return p;
}
static double N2108d28542(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 1;
} else if (((Double) i[3]).doubleValue() <= 60000.0) {
p = 1;
} else if (((Double) i[3]).doubleValue() > 60000.0) {
```

```
p = WekaClassifier.N4a352ab443(i);
}
return p;
}
static double N4a352ab443(Object []i) {
double p = Double.NaN;
if (i[4] == null) {
p = 0;
} else if (((Double) i[4]).doubleValue() <= 0.0) {
p = 0;
} else if (((Double) i[4]).doubleValue() > 0.0) {
p = 1;
}
return p;
}
static double N6886b57b44(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 0;
} else if (((Double) i[3]).doubleValue() <= 60000.0) {
p = 0;
```

```
} else if (((Double) i[3]).doubleValue() > 60000.0) {
p = WekaClassifier.N63f4d3d945(i);
}
return p;
}
static double N63f4d3d945(Object []i) {
double p = Double.NaN;
if (i[2] == null) {
p = 0;
} else if (i[2].equals("Female")) {
p = WekaClassifier.N5e807ed346(i);
} else if (i[2].equals("Male")) {
p = 1;
}
return p;
}
static double N5e807ed346(Object []i) {
double p = Double.NaN;
if (i[4] == null) {
p = 0;
} else if (((Double) i[4]).doubleValue() <= 0.0) {
```

```
p = WekaClassifier.N46067f1d47(i);
} else if (((Double) i[4]).doubleValue() > 0.0) {
p = 1;
}
return p;
}
static double N46067f1d47(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 1;
} else if (((Double) i[11]).doubleValue() <= 41.0) {
p = 1;
} else if (((Double) i[11]).doubleValue() > 41.0) {
p = 0;
}
return p;
}
static double N54836cd548(Object []i) {
double p = Double.NaN;
if (i[4] == null) {
p = 1;
```

```
} else if (((Double) i[4]).doubleValue() <= 1.0) {
p = 1;
} else if (((Double) i[4]).doubleValue() > 1.0) {
p = WekaClassifier.N5e3f4c49(i);
}
return p;
}
static double N5e3f4c49(Object []i) {
double p = Double.NaN;
if (i[2] == null) {
p = 0;
} else if (i[2].equals("Female")) {
p = 0;
} else if (i[2].equals("Male")) {
p = 1;
}
return p;
}
static double N2b8d96c450(Object []i) {
double p = Double.NaN;
if (i[6] == null) {
```

```
p = 1;
} else if (i[6].equals(”Clerical”)) {
p = WekaClassifier.N5ca5343251(i);
} else if (i[6].equals(”Professional”)) {
p = WekaClassifier.N73552c7453(i);
} else if (i[6].equals(”Manual”)) {
p = WekaClassifier.N6dcace4f65(i);
} else if (i[6].equals(”Skilled Manual”)) {
p = WekaClassifier.N1daedcd966(i);
} else if (i[6].equals(”Management”)) {
p = WekaClassifier.N432539c867(i);
}
return p;
}
static double N5ca5343251(Object []i) {
double p = Double.NaN;
if (i[10] == null) {
p = 1;
} else if (i[10].equals(”Europe”)) {
p = WekaClassifier.N6205ee2252(i);
} else if (i[10].equals(”Pacific”)) {
```

```
p = 1;
} else if (i[10].equals("North America")) {
p = 0;
}
return p;
}
static double N6205ee2252(Object []i) {
double p = Double.NaN;
if (i[4] == null) {
p = 1;
} else if (((Double) i[4]).doubleValue() <= 0.0) {
p = 1;
} else if (((Double) i[4]).doubleValue() > 0.0) {
p = 0;
}
return p;
}
static double N73552c7453(Object []i) {
double p = Double.NaN;
if (i[4] == null) {
p = 1;
```

```
} else if (((Double) i[4]).doubleValue() <= 0.0) {
p = WekaClassifier.N5cd9aed354(i);
} else if (((Double) i[4]).doubleValue() > 0.0) {
p = WekaClassifier.N576d273963(i);
}
return p;
}
static double N5cd9aed354(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 1;
} else if (((Double) i[3]).doubleValue() <= 60000.0) {
p = WekaClassifier.N5764ce4455(i);
} else if (((Double) i[3]).doubleValue() > 60000.0) {
p = 0;
}
return p;
}
static double N5764ce4455(Object []i) {
double p = Double.NaN;
if (i[7] == null) {
```

```
p = 1;
} else if (i[7].equals("Yes")) {
p = WekaClassifier.N1b21ff2f56(i);
} else if (i[7].equals("No")) {
p = 1;
}
return p;
}
static double N1b21ff2f56(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
p = 1;
} else if (((Double) i[8]).doubleValue() <= 0.0) {
p = WekaClassifier.N54f8749357(i);
} else if (((Double) i[8]).doubleValue() > 0.0) {
p = WekaClassifier.N2a36d9fe60(i);
}
return p;
}
static double N54f8749357(Object []i) {
double p = Double.NaN;
```

```
if (i[2] == null) {
p = 1;
} else if (i[2].equals(”Female”)) {
p = WekaClassifier.N3ad8a7a58(i);
} else if (i[2].equals(”Male”)) {
p = 1;
}
return p;
}
static double N3ad8a7a58(Object []i) {
double p = Double.NaN;
if (i[1] == null) {
p = 0;
} else if (i[1].equals(”Married”)) {
p = WekaClassifier.N765e517859(i);
} else if (i[1].equals(”Single”)) {
p = 1;
}
return p;
}
static double N765e517859(Object []i) {
```

```
double p = Double.NaN;
if (i[11] == null) {
p = 0;
} else if (((Double) i[11]).doubleValue() <= 38.0) {
p = 0;
} else if (((Double) i[11]).doubleValue() > 38.0) {
p = 1;
}
return p;
}
static double N2a36d9fe60(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 1;
} else if (((Double) i[0]).doubleValue() <= 19465.0) {
p = WekaClassifier.N7721c85861(i);
} else if (((Double) i[0]).doubleValue() > 19465.0) {
p = 0;
}
return p;
}
```

```
static double N7721c85861(Object []i) {
double p = Double.NaN;
if (i[1] == null) {
p = 0;
} else if (i[1].equals("Married")) {
p = WekaClassifier.N4ae8468262(i);
} else if (i[1].equals("Single")) {
p = 1;
}
return p;
}
static double N4ae8468262(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 1;
} else if (((Double) i[0]).doubleValue() <= 16571.0) {
p = 1;
} else if (((Double) i[0]).doubleValue() > 16571.0) {
p = 0;
}
return p;
```

```
}
static double N576d273963(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
p = 0;
} else if (((Double) i[8]).doubleValue() <= 0.0) {
p = WekaClassifier.N3f2a3c0664(i);
} else if (((Double) i[8]).doubleValue() > 0.0) {
p = 1;
}
return p;
}
static double N3f2a3c0664(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 0;
} else if (((Double) i[3]).doubleValue() <= 80000.0) {
p = 0;
} else if (((Double) i[3]).doubleValue() > 80000.0) {
p = 1;
}
```

```
return p;
}
static double N6dcace4f65(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 1;
} else if (((Double) i[0]).doubleValue() <= 20863.0) {
p = 1;
} else if (((Double) i[0]).doubleValue() > 20863.0) {
p = 0;
}
return p;
}
static double N1daedcd966(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 1;
} else if (((Double) i[11]).doubleValue() <= 35.0) {
p = 1;
} else if (((Double) i[11]).doubleValue() > 35.0) {
p = 0;
```

```
}
return p;
}
static double N432539c867(Object []i) {
double p = Double.NaN;
if (i[4] == null) {
p = 0;
} else if (((Double) i[4]).doubleValue() <= 1.0) {
p = WekaClassifier.N40f49be868(i);
} else if (((Double) i[4]).doubleValue() > 1.0) {
p = 1;
}
return p;
}
static double N40f49be868(Object []i) {
double p = Double.NaN;
if (i[10] == null) {
p = 0;
} else if (i[10].equals("Europe")) {
p = 0;
} else if (i[10].equals("Pacific")) {
```

```
p = 0;
} else if (i[10].equals("North America")) {
p = WekaClassifier.N7b27bda869(i);
}
return p;
}
static double N7b27bda869(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 0;
} else if (((Double) i[3]).doubleValue() <= 60000.0) {
p = 0;
} else if (((Double) i[3]).doubleValue() > 60000.0) {
p = 1;
}
return p;
}
static double N6e149fc270(Object []i) {
double p = Double.NaN;
if (i[5] == null) {
p = 1;
```

```
} else if (i[5].equals(”Graduate Degree”)) {

p = 1;

} else if (i[5].equals(”Partial College”)) {

p = WekaClassifier.Nc04b20471(i);

} else if (i[5].equals(”High School”)) {

p = 0;

} else if (i[5].equals(”Bachelors”)) {

p = WekaClassifier.N218659d874(i);

} else if (i[5].equals(”Partial High School”)) {

p = 1;

}

return p;

}

static double Nc04b20471(Object []i) {

double p = Double.NaN;

if (i[1] == null) {

p = 1;

} else if (i[1].equals(”Married”)) {

p = WekaClassifier.N8432ac672(i);

} else if (i[1].equals(”Single”)) {

p = 0;
```

```
}
return p;
}
static double N8432ac672(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 1;
} else if (((Double) i[3]).doubleValue() <= 50000.0) {
p = WekaClassifier.N71ad316673(i);
} else if (((Double) i[3]).doubleValue() > 50000.0) {
p = 1;
}
return p;
}
static double N71ad316673(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 1;
} else if (((Double) i[11]).doubleValue() <= 50.0) {
p = 1;
} else if (((Double) i[11]).doubleValue() > 50.0) {
```

```
p = 0;
}
return p;
}
static double N218659d874(Object []i) {
double p = Double.NaN;
if (i[7] == null) {
p = 0;
} else if (i[7].equals(”Yes”)) {
p = WekaClassifier.N7b74851975(i);
} else if (i[7].equals(”No”)) {
p = 1;
}
return p;
}
static double N7b74851975(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 0;
} else if (((Double) i[11]).doubleValue() <= 43.0) {
p = 0;
```

```
} else if (((Double) i[11]).doubleValue() > 43.0) {
p = 1;
}
return p;
}
static double N1ece6ba176(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 1;
} else if (((Double) i[3]).doubleValue() <= 20000.0) {
p = WekaClassifier.N1fc26d477(i);
} else if (((Double) i[3]).doubleValue() > 20000.0) {
p = 1;
}
return p;
}
static double N1fc26d477(Object []i) {
double p = Double.NaN;
if (i[7] == null) {
p = 1;
} else if (i[7].equals("Yes")) {
```

```
p = WekaClassifier.N5159295078(i);
} else if (i[7].equals(”No”)) {
p = 1;
}
return p;
}
static double N5159295078(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 1;
} else if (((Double) i[0]).doubleValue() <= 26821.0) {
p = WekaClassifier.N34a76ce979(i);
} else if (((Double) i[0]).doubleValue() > 26821.0) {
p = 0;
}
return p;
}
static double N34a76ce979(Object []i) {
double p = Double.NaN;
if (i[4] == null) {
p = 1;
```

```
} else if (((Double) i[4]).doubleValue() <= 3.0) {
p = 1;
} else if (((Double) i[4]).doubleValue() > 3.0) {
p = WekaClassifier.N5bdf500080(i);
}
return p;
}
static double N5bdf500080(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
p = 1;
} else if (((Double) i[8]).doubleValue() <= 0.0) {
p = 1;
} else if (((Double) i[8]).doubleValue() > 0.0) {
p = 0;
}
return p;
}
static double N21fd5f1f81(Object []i) {
double p = Double.NaN;
if (i[10] == null) {
```

```
p = 1;
} else if (i[10].equals(”Europe”)) {
p = 1;
} else if (i[10].equals(”Pacific”)) {
p = WekaClassifier.N16ffda882(i);
} else if (i[10].equals(”North America”)) {
p = WekaClassifier.N378928db97(i);
}
return p;
}
static double N16ffda882(Object []i) {
double p = Double.NaN;
if (i[1] == null) {
p = 1;
} else if (i[1].equals(”Married”)) {
p = 1;
} else if (i[1].equals(”Single”)) {
p = WekaClassifier.N5f76215483(i);
}
return p;
}
```

```
static double N5f76215483(Object []i) {
double p = Double.NaN;
if (i[5] == null) {
p = 1;
} else if (i[5].equals("Graduate Degree")) {
p = 1;
} else if (i[5].equals("Partial College")) {
p = WekaClassifier.N448642c284(i);
} else if (i[5].equals("High School")) {
p = WekaClassifier.N4a88f1b385(i);
} else if (i[5].equals("Bachelors")) {
p = WekaClassifier.N354a659590(i);
} else if (i[5].equals("Partial High School")) {
p = WekaClassifier.N35c95f3396(i);
}
return p;
}
static double N448642c284(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 1;
```

```
} else if (((Double) i[3]).doubleValue() <= 70000.0) {
p = 1;
} else if (((Double) i[3]).doubleValue() > 70000.0) {
p = 0;
}
return p;
}
static double N4a88f1b385(Object []i) {
double p = Double.NaN;
if (i[3] == null) {
p = 1;
} else if (((Double) i[3]).doubleValue() <= 20000.0) {
p = 1;
} else if (((Double) i[3]).doubleValue() > 20000.0) {
p = WekaClassifier.N64b444ff86(i);
}
return p;
}
static double N64b444ff86(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
```

```
p = 0;
} else if (((Double) i[8]).doubleValue() <= 2.0) {
p = WekaClassifier.N771df4ff87(i);
} else if (((Double) i[8]).doubleValue() > 2.0) {
p = 1;
}
return p;
}
static double N771df4ff87(Object []i) {
double p = Double.NaN;
if (i[7] == null) {
p = 0;
} else if (i[7].equals("Yes")) {
p = WekaClassifier.N4fbfce7188(i);
} else if (i[7].equals("No")) {
p = 0;
}
return p;
}
static double N4fbfce7188(Object []i) {
double p = Double.NaN;
```

```
if (i[11] == null) {
p = 0;
} else if (((Double) i[11]).doubleValue() <= 49.0) {
p = 0;
} else if (((Double) i[11]).doubleValue() > 49.0) {
p = WekaClassifier.N398a859e89(i);
}
return p;
}
static double N398a859e89(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 1;
} else if (((Double) i[0]).doubleValue() <= 19441.0) {
p = 1;
} else if (((Double) i[0]).doubleValue() > 19441.0) {
p = 0;
}
return p;
}
static double N354a659590(Object []i) {
```

```
double p = Double.NaN;
if (i[0] == null) {
p = 1;
} else if (((Double) i[0]).doubleValue() <= 24586.0) {
p = WekaClassifier.N2753348891(i);
} else if (((Double) i[0]).doubleValue() > 24586.0) {
p = 1;
}
return p;
}
static double N2753348891(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
p = 1;
} else if (((Double) i[8]).doubleValue() <= 2.0) {
p = WekaClassifier.N479de0e392(i);
} else if (((Double) i[8]).doubleValue() > 2.0) {
p = WekaClassifier.N67c1bcbd94(i);
}
return p;
}
```

```
static double N479de0e392(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 0;
} else if (((Double) i[11]).doubleValue() <= 65.0) {
p = WekaClassifier.N4e177bd093(i);
} else if (((Double) i[11]).doubleValue() > 65.0) {
p = 1;
}
return p;
}
static double N4e177bd093(Object []i) {
double p = Double.NaN;
if (i[4] == null) {
p = 0;
} else if (((Double) i[4]).doubleValue() <= 0.0) {
p = 0;
} else if (((Double) i[4]).doubleValue() > 0.0) {
p = 1;
}
return p;
```

```
}
static double N67c1bcbd94(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 0;
} else if (((Double) i[0]).doubleValue() <= 13151.0) {
p = WekaClassifier.N604e518295(i);
} else if (((Double) i[0]).doubleValue() > 13151.0) {
p = 1;
}
return p;
}
static double N604e518295(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 1;
} else if (((Double) i[11]).doubleValue() <= 31.0) {
p = 1;
} else if (((Double) i[11]).doubleValue() > 31.0) {
p = 0;
}
```

```
return p;
}
static double N35c95f3396(Object []i) {
double p = Double.NaN;
if (i[6] == null) {
p = 1;
} else if (i[6].equals("Clerical")) {
p = 1;
} else if (i[6].equals("Professional")) {
p = 0;
} else if (i[6].equals("Manual")) {
p = 1;
} else if (i[6].equals("Skilled Manual")) {
p = 1;
} else if (i[6].equals("Management")) {
p = 1;
}
return p;
}
static double N378928db97(Object []i) {
double p = Double.NaN;
```

```
if (i[1] == null) {
p = 1;
} else if (i[1].equals(”Married”)) {
p = 1;
} else if (i[1].equals(”Single”)) {
p = WekaClassifier.Ndd95e5998(i);
}
return p;
}
static double Ndd95e5998(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
p = 1;
} else if (((Double) i[8]).doubleValue() <= 2.0) {
p = 1;
} else if (((Double) i[8]).doubleValue() > 2.0) {
p = WekaClassifier.N39c12c2999(i);
}
return p;
}
static double N39c12c2999(Object []i) {
```

```
double p = Double.NaN;
if (i[7] == null) {
p = 0;
} else if (i[7].equals(”Yes”)) {
p = WekaClassifier.N393a555e100(i);
} else if (i[7].equals(”No”)) {
p = WekaClassifier.N3020c376105(i);
}
return p;
}
static double N393a555e100(Object []i) {
double p = Double.NaN;
if (i[6] == null) {
p = 1;
} else if (i[6].equals(”Clerical”)) {
p = 1;
} else if (i[6].equals(”Professional”)) {
p = 0;
} else if (i[6].equals(”Manual”)) {
p = 0;
} else if (i[6].equals(”Skilled Manual”)) {
```

```
p = WekaClassifier.N24baabac101(i);
} else if (i[6].equals("Management")) {
p = WekaClassifier.N253d6362103(i);
}
return p;
}
static double N24baabac101(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 1;
} else if (((Double) i[0]).doubleValue() <= 22102.0) {
p = WekaClassifier.N5b5cbc0a102(i);
} else if (((Double) i[0]).doubleValue() > 22102.0) {
p = 0;
}
return p;
}
static double N5b5cbc0a102(Object []i) {
double p = Double.NaN;
if (i[4] == null) {
p = 1;
```

```
} else if (((Double) i[4]).doubleValue() <= 3.0) {
p = 1;
} else if (((Double) i[4]).doubleValue() > 3.0) {
p = 0;
}
return p;
}
static double N253d6362103(Object []i) {
double p = Double.NaN;
if (i[5] == null) {
p = 1;
} else if (i[5].equals("Graduate Degree")) {
p = 1;
} else if (i[5].equals("Partial College")) {
p = 0;
} else if (i[5].equals("High School")) {
p = 0;
} else if (i[5].equals("Bachelors")) {
p = WekaClassifier.N6147ca07104(i);
} else if (i[5].equals("Partial High School")) {
p = 0;
```

```
}
return p;
}
static double N6147ca07104(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 1;
} else if (((Double) i[11]).doubleValue() <= 40.0) {
p = 1;
} else if (((Double) i[11]).doubleValue() > 40.0) {
p = 0;
}
return p;
}
static double N3020c376105(Object []i) {
double p = Double.NaN;
if (i[11] == null) {
p = 1;
} else if (((Double) i[11]).doubleValue() <= 44.0) {
p = 1;
} else if (((Double) i[11]).doubleValue() > 44.0) {
```

```
p = WekaClassifier.N370090a9106(i);
}
return p;
}
static double N370090a9106(Object []i) {
double p = Double.NaN;
if (i[8] == null) {
p = 1;
} else if (((Double) i[8]).doubleValue() <= 3.0) {
p = WekaClassifier.N6196375107(i);
} else if (((Double) i[8]).doubleValue() > 3.0) {
p = 1;
}
return p;
}
static double N6196375107(Object []i) {
double p = Double.NaN;
if (i[0] == null) {
p = 0;
} else if (((Double) i[0]).doubleValue() <= 21322.0) {
p = 0;
```

```
} else if (((Double) i[0]).doubleValue() > 21322.0) {
p = 1;
}
return p;
}
}
```